

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

О. Є. Поморцева

ПРОГРАМУВАННЯ
ГЕОІНФОРМАЦІЙНИХ ЗАДАЧ
ЛАБОРАТОРНИЙ ПРАКТИКУМ

НАВЧАЛЬНИЙ ПОСІБНИК

ХАРКІВ
ХНУМГ ім. О. М. Бекетова
2015

УДК 004.42:004.451.42(076)

ББК 32.973-018.2я73

П55

Автор

Поморцева Олена Євгенівна, к.т.н., доц.

Рецензенти:

Янцевич А. А., докт. фіз.-мат. наук, професор кафедри математичних методів в економіці ХНУ ім. В. Н. Каразіна;

Колгатін О. Г., докт. пед. наук, професор кафедри інформатики ХНПУ імені Г. С. Сковороди

Рекомендовано на засіданні

*Вченої ради ХНУМГ ім. О. М. Бекетова,
протокол № 10 від 24 квітня 2015 р.*

Поморцева О. Є.

П55 Програмування геоінформаційних задач: лабораторний практикум: навч. посібник / О. Є. Поморцева; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2015. – 111 с.

Матеріал посібника викладено на прикладі завдань геоінформаційного спрямування, які вирішуються за допомогою програмування, що дозволяє істотно автоматизувати роботу користувачів геоінформаційних систем. Засвоєння матеріалу спрямоване на оволодіння інструментальними засобами, які дають можливість розробляти самостійні проекти у середовищі ArcMap. Досягнутий рівень компетентності дозволить ефективно використовувати можливості мови програмування Visual Basic for Applications під час розв'язання завдань професійної діяльності для студентів напряму підготовки 6.080101 – Геодезія, картографія та землеустрій, створить основу для самостійного освоєння нових мов програмування.

УДК 004.42:004.451.42(076)

ББК 32.973-018.2я73

© О. Є. Поморцева, 2015

© ХНУМГ ім. О. М. Бекетова, 2015

ЗМІСТ

ВСТУП.....	5
ЛАБОРАТОРНА РОБОТА № 1 ЗНАЙОМСТВО З ІНТЕГРОВАНИМ СЕРЕДОВИЩЕМ РЕДАКТОРА VBA	8
Підготовка до роботи.....	8
ТЕОРЕТИЧНІ ВІДОМОСТІ	8
Хід роботи.....	9
Завдання для САМОСТІЙНОГО ВИКОНАННЯ	17
ЛАБОРАТОРНА РОБОТА № 2 ЛІНІЙНИЙ ОБЧИСЛЮВАЛЬНИЙ ПРОЦЕС. СТВОРЕННЯ ВЛАСНИХ КНОПОК.....	18
Підготовка до роботи.....	18
ТЕОРЕТИЧНІ ВІДОМОСТІ	18
Хід роботи.....	20
Завдання для САМОСТІЙНОГО ВИКОНАННЯ	31
Контрольні питання	33
ЛАБОРАТОРНА РОБОТА № 3 НАЛАШТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА.....	34
Підготовка до роботи.....	34
ТЕОРЕТИЧНІ ВІДОМОСТІ	35
Хід роботи.....	36
Завдання для САМОСТІЙНОГО ВИКОНАННЯ	40
Контрольні питання	41
ЛАБОРАТОРНА РОБОТА № 4 ПРОГРАМУВАННЯ ОБ'ЄКТІВ ФОРМИ. РОЗГАЛУЖЕНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ.....	42
Підготовка до роботи.....	42
ТЕОРЕТИЧНІ ВІДОМОСТІ	42
Хід роботи.....	45
Завдання для САМОСТІЙНОГО ВИКОНАННЯ	51
Контрольні питання	51
ЛАБОРАТОРНА РОБОТА № 5 ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ.....	51
Підготовка до роботи.....	52
ТЕОРЕТИЧНІ ВІДОМОСТІ	52
Хід роботи.....	54
Завдання для САМОСТІЙНОГО ВИКОНАННЯ	64
Контрольні питання	64

ЛАБОРАТОРНА РОБОТА № 6 ВИКОРИСТАННЯ ПІДПРОГРАМ ТА ФУНКЦІЙ.....	65
Підготовка до роботи.....	65
ТЕОРЕТИЧНІ ВІДОМОСТІ	65
Хід роботи.....	67
Завдання для самостійного виконання	81
Контрольні питання	81
ЛАБОРАТОРНА РОБОТА № 7 РОЗРОБКА ВЛАСНИХ ОБ'ЄКТІВ... 82	82
Підготовка до роботи.....	82
ТЕОРЕТИЧНІ ВІДОМОСТІ	82
Хід роботи.....	84
Завдання для самостійного виконання	91
Контрольні питання	91
ЛАБОРАТОРНА РОБОТА № 8 АВТОМАТИЗАЦІЯ РОБОТИ В ARСМАР	92
Підготовка до роботи.....	92
ТЕОРЕТИЧНІ ВІДОМОСТІ	92
Хід роботи.....	93
Завдання для самостійного виконання	99
Контрольні питання	99
РОЗРАХУНКОВО-ГРАФІЧНЕ ЗАВДАННЯ.	
РОЗРОБКА ГІС ПРОЕКТУ У СЕРЕДОВИЩІ ARСМАР ДЛЯ ВИРІШЕННЯ ПРИКЛАДНИХ ГІС ЗАДАЧ.....	100
Загальні вимоги до оформлення РГЗ	101
Варіанти завдань	102
ДОДАТКИ.....	108
РЕКОМЕНДОВАНІ ДЖЕРЕЛА.....	111

ВСТУП

Навчальний посібник підготовлено згідно з програмою вибіркової навчальної дисципліни **«Програмування геоінформаційних задач»**, яка має професійне спрямування для підготовки бакалаврів за напрямом «Геодезія, картографія та землеустрій».

До лабораторного практикуму входять такі роботи:

- Знайомство з інтегрованим середовищем редактора VBA.
- Лінійний обчислювальний процес. Створення власних кнопок.
- Налаштування інтерфейсу користувача.
- Програмування об'єктів форми. Розгалужені обчислювальні процеси.
- Циклічні обчислювальні процеси.
- Використання підпрограм та функцій.
- Розробка власних об'єктів.
- Автоматизація роботи в ArcMap.

У даному навчальному посібнику розглянуто основні теоретичні поняття та терміни, які допомагають оволодіти основними принципами програмування за допомогою мови **Visual Basic for Applications (VBA)** – спрощеної версії Visual Basic, прийоми і засоби створення форм та програмування їх об'єктів у програмному забезпеченні ArcMap.

Кожна лабораторна робота передбачає попередню підготовку до неї. Перед виконанням вправи необхідно спочаку відповісти на низку питань, для цього кінці кожної лабораторної роботи наведено завдання для самостійного виконання та контрольні питання, що значно покращує засвоєння викладеного матеріалу. Слід також підкреслити що завдання для самостійного виконання поділено на дві категорії –загального та ускладненого рівня, що стимулює учнів до отримання додаткових балів.

Професійні компетентності, що формуються під час вивчення навчальної дисципліни «Програмування геоінформаційних задач».

В процесі навчання студенти отримують необхідні знання під час лекційних занять, закріплюють і поглиблюють їх, набуваючи при цьому практичні навички та вміння при виконанні лабораторних робіт. Особливе значення має індивідуальна робота студентів, під час виконання якої вони самостійно розробляють проект у середовищі ArcMap для вирішення прикладних геоінформаційних (ГІС) задач. В процесі роботи над проектом студенти набувають навичок опрацювання науково-технічної літератури,

навчаються самостійно приймати рішення та робити висновки. В результаті засвоєння матеріалу навчального посібника у студентів повинні сформуватися наступні компетентності.

Проектні, що пов'язані з використанням основ програмування мовою VBA, знанням основних принципів створення додатків.

Технічні, пов'язані із застосуванням засобів програмування у прикладних пакетах для комплексної обробки існуючих даних при розв'язанні прикладних геоінформаційних задач.

Аналітичні, що пов'язані з використанням прикладних пакетів для аналізу даних предметної області у зазначені терміни з використанням програмування засобами персональних комп'ютерів, самостійного вибору й освоєння нових програмних продуктів.

Принципи, які покладено в основу побудови навчального посібника.

В основу побудови посібника покладено принципи системності і практичної спрямованості з використанням сучасної мови програмування — Visual Basic.

Принцип практичної спрямованості передбачає фундаментальну наукову підготовку й активне практичне навчання студентів. Посібник створює умови для формування навичок і вмінь за допомогою практичної форми навчання.

У роботі розглядається англomовна версія MS Visual Basic, тому назви елементів управління, меню, команди і відповідні їм діалогові вікна наводяться англійською мовою.

У посібнику зміст практичного навчання реалізовано на базі методу активної рефлексії. Застосування цього методу припускає вивчення й обмірковане повторення студентами операцій, які необхідні для досягнення поставленої мети.

Під час викладення матеріалу дається короткий опис теоретичних основ, аналіз засобів застосування отриманих знань та детально розглядається процес розробки додатків засобами Visual Basic for Applications. Завдяки цьому студенти навчаються створювати форми з подальшим програмуванням цих об'єктів самостійно з використанням матеріалу посібника.

Важливим є те, що у практичній частині посібника використовуються завдання, які мають практичну спрямованість за напрямом підготовки. Формування навичок може здійснюватися як під керівництвом викладача в

аудиторії, так і вдома шляхом самостійного створення індивідуальних проєктів на основі розглянутої методики.

Під час вибору матеріалу враховувалися обмеження, які накладає на навчальний процес час навчання. До посібника включено мінімальний набір засобів VBA, які необхідні для створення власних форм з подальшим програмуванням цих об'єктів.

У ході викладення матеріалу використовуються **навігаційні підказки** у вигляді позначок, які допоможуть зорієнтуватися в структурі навчального посібника:

- **напівжирне написання** – терміни програми Visual Basic;
- ***курсивне напівжирне написання*** – назви, які вводить студент.
- **ПРИМІТКА** – роз'яснення, за допомогою якого можна вирішити поставлене завдання;
- **1*** – питання підвищеної складності (у завданнях для самостійного виконання).

Для визначення рівня засвоєння матеріалу кожного підрозділу пропонуються контрольні питання. Паралельно зі всіма лабораторними роботами студенти мають працювати над своїм індивідуальним завданням згідно свого варіанту. Варіанти індивідуальних завдань наведені у відповідному розділі навчального посібника.

Засвоєння у повному обсязі матеріалу даного навчального посібника допоможе розвинути здатність до подальшого навчання, самостійного розвитку та оволодіння інструментальними засобами для розв'язання завдань за допомогою програмування під час розв'язання прикладних задач геоінформаційного спрямування.

Посібник апробований під час викладання дисципліни «Програмування геоінформаційних задач» студентам Харківського національного університету міського господарства імені О. М. Бекетова.

ЛАБОРАТОРНА РОБОТА № 1

ЗНАЙОМСТВО З ІНТЕГРОВАНИМ СЕРЕДОВИЩЕМ РЕДАКТОРА VBA

Мета: виробити уміння і навички роботи з компонентами інтегрованого середовища розробки додатків.

Призначення: виконавши роботу, ви навчитесь створювати додатки за допомогою об'єктно-орієнтованої мови програмування Visual Basic, яка вбудована у ArcMap Desktop.

Підготовка до роботи

1. У чому полягають особливості мови Visual Basic for Applications?
2. Поясніть що таке «код» програми.
3. Поясніть що таке «процедура».
4. Поясніть що таке «подія».
5. Яким чином відкрити редактор Visual Basic у ArcMap?

Ключові слова

Об'єктно-орієнтована мова програмування, код, процедура, подія, оператор, коментар, макрос.

Теоретичні відомості

Visual Basic for Applications – спрощена версія Visual Basic, однієї з безліч об'єктно-орієнтованих мов програмування. Також ви можете використовувати інші мови для програмування ArcObjects (наприклад C++ або VisualBasic), але VBA є вбудованою в ArcGis Desktop і завантажується разом з ArcMap або ArcCatalog.

Головна відмінність між VBA та іншими об'єктно-орієнтованими мовами програмування в тому, що VBA створена спеціально для роботи з додатками, до складу яких вона входить. Програмісти, які працюють з такими мовами програмування як C++ як правило, будують додатки на порожньому місці, працюючи з низкою об'єктів, тоді як програмісти VBA набувають додатки, такі як ArcMap, до складу яких входить VBA. Програмісти VBA можуть використовувати вбудовану функціональність додатку, в якому вони працюють.

У VBA є свій власний набір засобів розробки. Наприклад, в цій мові є вікна для організації й зберігання коду, який Ви пишете; інструменти для того, щоб створити діалогові вікна і їхні компоненти (кнопки, випадаючі

списки, і так далі); інструменти для того, щоб налаштувати код.

Код організований в процедури. Процедура включає усі інструкції, при дотриманні яких повинна бути виконана деяка задача, така як наприклад друк карти чи якась інша. Процедури можуть бути пов'язані між собою так, щоб коли одна з них закінчується, повинна починатися наступна.

У Вас може іноді бути ланцюжок процедур, які виконуються одна за одною, але перша процедура в ланцюгу вимагає людського втручання. Такі дії як відкриття додатку, клацання кнопкою, або переміщення покажчика миші називають «подіями». **Подія** – це те, що запускає процедуру на виконання.

Хід роботи

Вправа 1

Знайомство з інтегрованим середовищем редактора VBA

- Відкрийте додаток ArcGis / ArcMap.
- Відкрийте вікно інтегрованого середовища розробки додатків редактора Visual Basic натисненням комбінації клавіш **Alt + F11** або за допомогою меню **Tools / Macros / Visual Basic Editor**.
- Ознайомтеся зі структурою головного вікна середовища редактора Visual Basic (рис. 1.1).

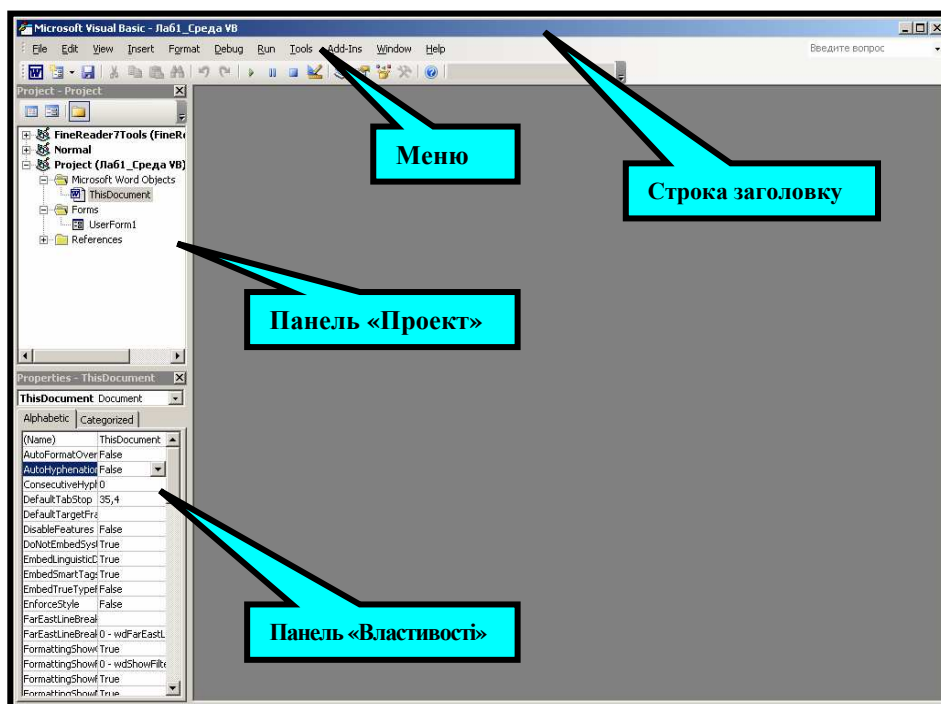




Рисунок 1.1 – Зовнішній вигляд вікна редактора Visual Basic

- Видаліть, а потім відновіть вікно **Проект** (меню **View** команда **Project Explorer** ) і **Властивості** (меню **View** команда **Properties Window** ) .
- Додайте до проекту вікно форми, клацнувши кнопку **Insert UserForm**, яка розташована на стандартній панелі інструментів (рис. 1.2).

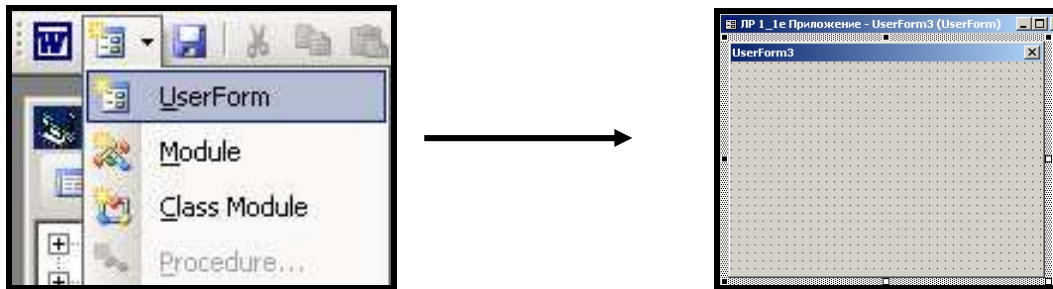


Рисунок 1.2 – Додавання до проекту форми

- Дізнайтеся значення властивостей форми **Name** та **Caption** (дивіться на панель **Properties**).
- Знайдіть на панелі елементів управління кнопки наступних елементів: **Label** та **CommandButton** і додайте їх на форму перетягнувши лівою кнопкою миші (рис. 1.3).

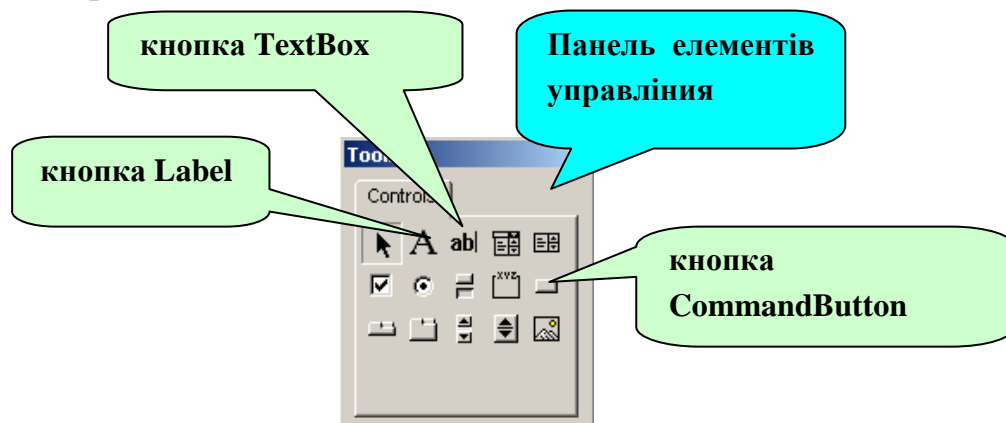




Рисунок 1.3 – Панель елементів управління (Toolbox)

- Подвійним клацанням миші на формі викличте вікно коду (рис. 1.4).
- За допомогою команд **View Code**  та **View Object**  вікна проекту перейдіть з вікна коду у вікно форми і назад (рис. 1.5).

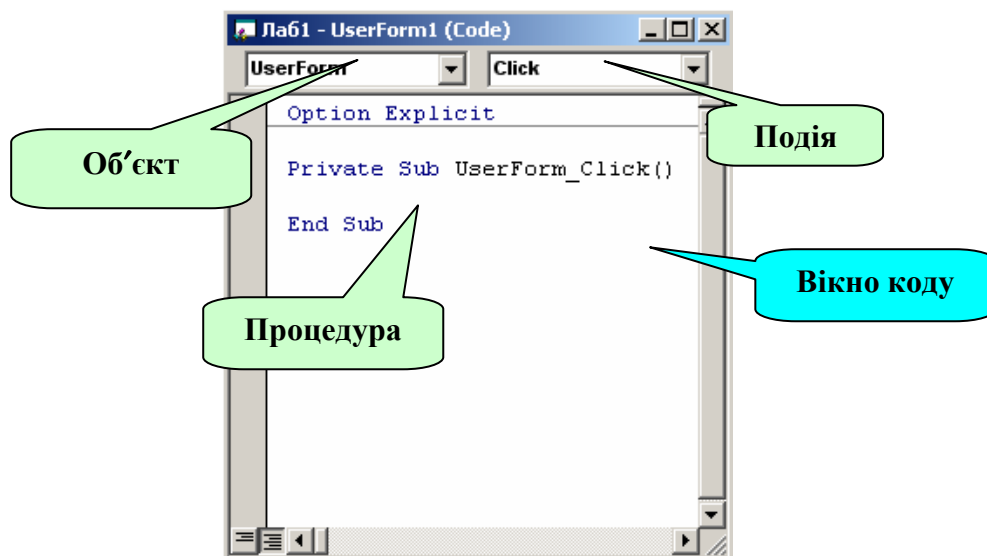


Рисунок 1.4 – Вікно коду

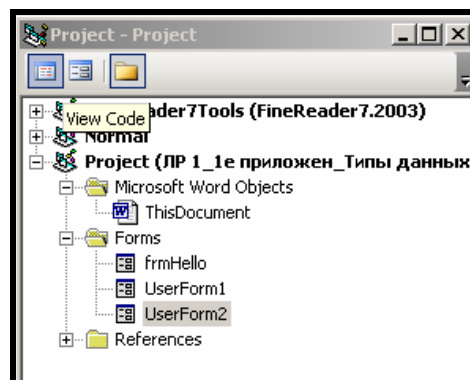



Рисунок 1.5 – Вікно проекту

- Запустіть програму на виконання, виконавши команду **Run Sub / UserForm**  а потім завершіть її роботу. Як змінився зовнішній вигляд форми?
- Збережіть документ ArcMap разом з програмою VBA (ім'я файла *Середовище VB.mxd*) і закрийте вікно ArcMap.
- Подвійним клацанням миші на значку документа *Середовище VB.mxd* відкрийте його, перейдіть у вікно інтегрованого середовища розробки додатків редактора Visual Basic і переконайтеся, що в ньому є форма UserForm.

Вправа 2

Ваш перший додаток

Після короткого знайомства з інтегрованим середовищем та його можливостями, настав час застосувати отримані знання на практиці і створити найпростіший додаток.

Додаток «Привіт геоінформаційникам!» створюється наступним чином:

- Змініть розміри вже створеної форми, перетягнувши її край маркером за допомогою миші. В остаточному варіанті вона повинна мати розміри близько 7 см завширшки та 5 см заввишки.

ПРИМІТКА. Форму можна переміщувати, перетягуючи її мишею у вікні макета.

- Виділіть форму, клацнувши на ній мишею. Про те, що виділена саме форма, а не один з розміщених на ній елементів, можна судити по змісту вікна властивостей. Якщо це вікно відсутнє на екрані, натисніть F4.

Задайте значення двох властивостей форми (табл. 1.1):

- **Caption** – *Мій перший додаток. Автор: Петров* (після слова «Автор» вкажіть своє прізвище)
- **Name** - *frmHello*.

Текст, написаний в якості властивості **Caption**, буде виведений у заголовку форми. За допомогою дуже важливої властивості **Name** ми будемо посилатися на форму у програмі. Ім'я *frmHello* набагато краще за ім'я **Form1**, яке присвоює VBA властивості **Name** за замовченням. Те ж стосується властивості **Caption** цього додатку, яка змінена на *Мій перший додаток. Автор: Петров*.

Таблиця 1.1 – Загальноприйняті префікси елементів управління

Тип об'єкту	Призначення	Префікс
Label	напис	lbl
TextBox	текстове поле	txt
CommandButton	кнопка	cmd
CheckBox	прапорець	chk
OptionButton	перемикач	opt
Frame	група (рамка)	fra
ListBox	список	lst
ComboBox	поле зі списком	cbo
Image	малюнок	img
PictureBox	графічний фрейм (вікно з малюнком)	pic
OLE Container	об'єкт OLE	ole
Form	форма	frm
Shape	фігура	shp
ShipButton	лічильник	spb
ScrollBar	полоса прокрутки	slb

Клацніть один раз на елементі управління **Label** (напис) у вікні **Toolbox** і перемістіть курсор миші на форму. Курсор миші змінить свій вигляд на (+). Встановіть хрестик в тому місці, де буде знаходитись верхній кут елементу. Натисніть ліву кнопку миші та утримуючи її, перетягніть курсор до того місця, де буде знаходитист правий нижній кут створюваного елементу. При цьому буде видно прямокутний контур. Відпустіть кнопку миші. На формі з'явиться прямокутна область, для якої задайте у **влас- тивості Name** значення **IblHello**, а у властивість **Caption** введіть *Привіт геоінформаційникам!*

У властивості **ForeColor** на вкладці **Palette** встановіть колір так, щоб у формі, яка проектується текст напису *Привіт геоінформаційникам!* відображався синім кольором на бірюзовому полі. Використайте для цього властивості **ForeColor** та **BackColor** у вікні **Properties** (рис. 1.6).

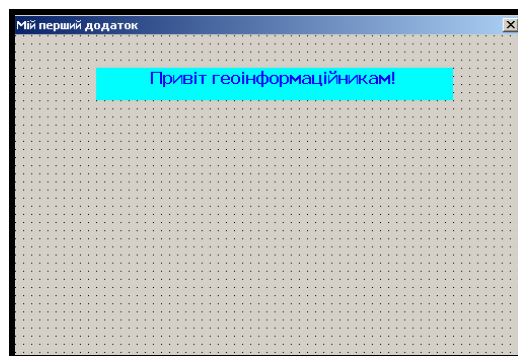


Рисунок 1.6 – Напис на формі

Потрібно передбачити, щоб під час виконання програми при клацан- ні на написі *Привіт геоінформаційникам!* колір змінювався на червоний.

Для цього двічі клацніть в області напису *Привіт геоінформаційни- кам!* Подвійне клацання на елементі (або формі) відкриває вікно програми зі стандартною подією. В даному випадку стандартна подія – **Click** (кла- цання мишею). На екрані повинен з'явитися шаблон процедури (або заго- товка).

Private Sub IblHello_Click()

End Sub

Не звертайте уваги на префікс **Private Sub**; зараз важливо лише ім'я процедури, **IblHello Click()** – воно означає, що після того, як користувач клацне мишею по вказаній області, буде виконаний код, що знаходиться в

цій процедури. Введіть у вільне поле даної процедури:

lblHello.ForeColor = vbRed

Таким чином, ви встановили червоний колір напису (рис. 1.7). Закрийте вікно діалогу з кодом програми.

ПРИМІТКА. Зверніть увагу на те, що при написанні коду після того, як ви поставили крапку після назви елемента форми – текстового напису **lblHello** з'явився випадаючий список з усіма діями, які можна застосувати до цього елемента.

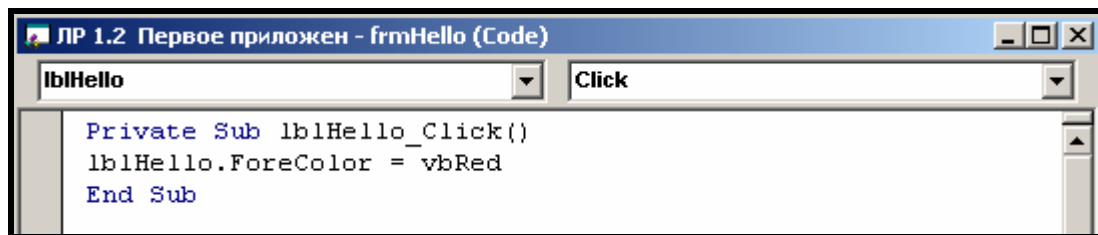


Рисунок 1.7 – Процедура зміни кольору напису на червоний

Створіть ще один елемент управління **Label**, для якого задайте у властивості **Name** ім'я **lblName**, а у властивість **Caption** введіть **Ваш Вася** (своє ім'я).

У властивості **ForeColor** у випадаючому списку на вкладці **Palette** вікна **Properties** встановіть такий колір, щоб в запроектованій формі текст напису **Ваш Вася** відображався зеленим кольором, а у властивості **BackColor** задайте фон – світложовтий (рис. 1.8).

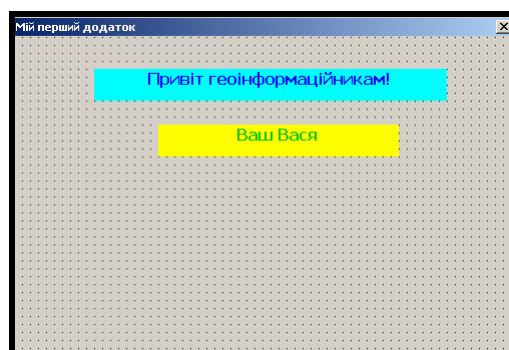


Рисунок 1.8 – Форма з двома написами

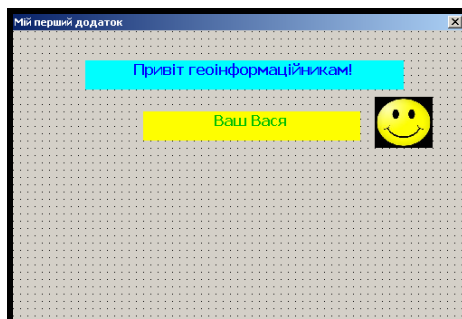
Потрібно передбачити, щоб при виконанні програми при клацанні по напису **Ваш Вася** колір змінювався на синій. Для цього виконайте дії, аналогічно описаним вище, але у вікні коду вкажіть – **lblName.ForeColor = vbBlue** (рис. 1.9).

```
Private Sub lblName_Click()
    lblName.ForeColor = vbBlue
End Sub
```

Рисунок 1.9 – Процедура зміни кольору напису на синій

Додійте малюнок на форму – елемент управління **Image**. Потім за допомогою властивості **Picture** вкажіть місцезнаходження бажаного малюнку. Розташуйте малюнок праворуч від напису **Ваш Вася**.

Передбачте, щоб під час виконання програми при клацанні по малюнку він зникав – в коді укажіть зміну властивості **Visible** (**True** або **False**) (рис. 1.10).



```
Private Sub ImgSmile_Click()
    ImgSmile.Visible = False
End Sub
```

Рисунок 1.10 – Форма з малюнком та код

Додайте на форму елемент-кнопку (**Command Button**) (рис. 1.11), перемістіть її в нижню центральну частину вікна форми і задайте наступні властивості:

- Властивість **Name** – **cmdEXIT**
- Властивість **Caption** – **Вихід**

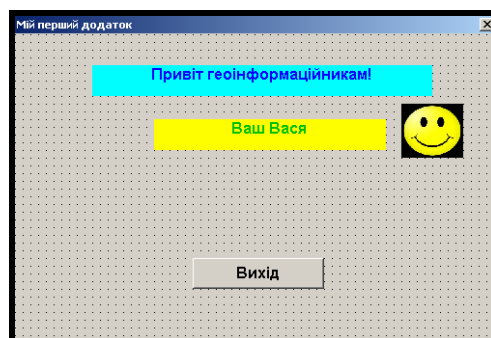



Рисунок 1.11 – Форма з кнопкою «Вихід»

Тепер двічі клацніть на кнопці **cmdEXIT** («Вихід»). Подвійне клацання на елементі (або формі) відкриває вікно програми зі стандартною подією. Для кнопки стандартною є подія **Click**. На екрані повинен з'явитися шаблон процедури (або заготовка) введіть між рядками **Private Sub** та **End Sub** наступний фрагмент програми (рис. 1.12):

```
Private Sub cmdEXIT_Click()  
Unload Me  
Set frmHello = Nothing  
End Sub
```


Рисунок 1.12 – Процедура завершення додатку

Коли користувач натискає («клацає») кнопку **cmdEXIT**, відбувається подія **cmdEXIT_Click**. В даному випадку вона повідомляє форму про те, що вона повинна вивантажити себе. Оскільки в нашому додатку немає інших форм, вивантаження форми призводить до завершення додатку (форми повинно бути привласнено ім'я **frmHello**).

Збережіть зміни натиснувши кнопку стандартної панелі інструментів **Save** (Зберегти) . При збереженні файлу зберігаються усі його форми. У нашому додатку є всього одна форма, якій відповідає один файл.

ПРИМІТКА. Зверніть увагу – наша форма описується трьома атрибутами: властивістю **Name** (**FrmHello**), властивістю **Caption** (**Мій перший додаток**) та ім'ям файлу (**Середовище VB**). Ви повинні чітко розуміти, чим відрізняються ці атрибути.

ПРИМІТКА. Властивість **Caption** виводиться в заголовок форми, властивість **Name** служить для роботи з формою у програмі, а ім'я файлу використовується операційною системою.

Виконайте команду **Run / Start** . Якщо все було зроблено правильно, на екрані з'являється форма з повідомленням ***Привіт геоінформаційникам!*** (рис. 1.13).

Завершіть роботу додатку натиснувши на кнопку **Вихід**. Якщо у вас щось не вийшло, повторіть описані дії та знайдіть помилку.

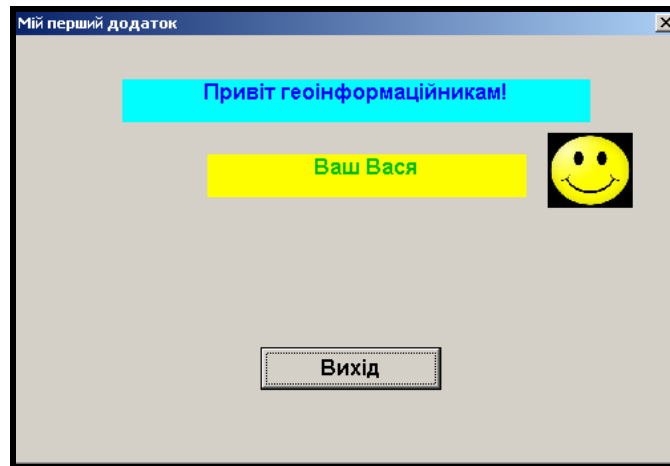


Рисунок 1.13 – Зовнішній вигляд спроектованої форми
«Мій перший додаток» після виконання програми

Ви тільки що створили додаток, зробили перший крок на шляху до своєї професійної майстерності у Visual Basic. На прикладі цієї найпростішої програми ви познайомилися з роботою Visual Basic. Хоча додаток ***Привіт геоінформаційникам!*** був дуже простим, він дає деяке уявлення про те, як працює Visual Basic.

Завдання для самостійного виконання

1. Передбачте, щоб під час виконання програми при повторному клацанні на тому місці, де був малюнок він знов з'являвся. (В коді змінюється значення властивості **Visible**).
2. Створіть **Label** «Дата» і передбачте, щоб при клацанні на тексті, з'являлася поточна дата.
3. * Передбачте, щоб під час відкриття документу з'являлася форма, створена в лабораторній роботі. (В процедурі обробки події **Open** для документу використовуйте метод **Show** форми).
4. Помістіть на кнопці «**Вихід**» малюнок замість напису.
5. Встановіть різні параметри тексту (шрифт, розмір, колір) для різних текстових полів на формі.
6. * Чи можна провести вивантаження форми без кнопки «**Вихід**»? Змініть для цього код.
7. Встановіть як фон форми, замість сірого кольору, фоновий малюнок.

ЛАБОРАТОРНА РОБОТА № 2

ЛІНІЙНИЙ ОБЧИСЛЮВАЛЬНИЙ ПРОЦЕС.

СТВОРЕННЯ ВЛАСНИХ КНОПОК

Мета: виробити уміння та навички роботи з панелями користувача і створення лінійних програм з подальшим підключенням їх до кнопок на панелях.

Призначення: виконавши роботу, ви навчитеся створювати панелі інструментів користувача та лінійні програми. А також підключати ці програми до кнопок на панелях інструментів.

Підготовка до роботи

1. Дайте визначення алгоритму.
2. Що таке алгоритмічна мова? Які мови ви знаєте?
3. Поясніть що таке «код» програми.
4. Яким чином створювати інтерфейс користувача у програмі ArcMap?
5. Яким чином підключати створений програмний код до кнопки на панелі інструментів у програмі ArcMap?

Ключові слова

Алгоритм, макрос, програмний код, панель інструментів, зображення (іконка) на кнопці, карта.

Теоретичні відомості

Розв'язання задач на ЕОМ (електронно-обчислювальна машина) – це процес перетворення початкових даних в кінцеві результати. Цей процес є обробкою даних і складається з певної послідовності дій.

Принцип розв'язання задач на ЕОМ полягає в представленні задачі у вигляді програми обробки даних і реалізації її на ЕОМ з метою отримання кінцевих результатів при заданих початкових даних. Розробка програми заснована на можливості представити будь-яку задачу у математичній формі: математичними формулами та іншими співвідношеннями, якими задаються правила обробки даних. Програма визначає послідовність реалізації цих правил на ЕОМ.

Алгоритм – це чітко задана послідовність дій, які повинні бути виконані для вирішення задачі. Алгоритм – це план роботи комп'ютера. Одну ціль можна досягти різними способами (алгоритмами).

Словесний опис алгоритму.

1. Введення
2. Обчислення
3. Виведення

Програма на алгоритмічній мові є формою представлення алгоритму. Вона включає опис алгоритму і даних на алгоритмічній мові. Опис даних – це вказівки на виділення певного об'єму пам'яті і форму представлення даних для їхнього збереження і обробки.

Алгоритмічна мова – програма для комп'ютера.

Таким чином, для вирішення задач на ЕОМ необхідно: розробити математичну модель задачі, визначити початкові дані і кінцеві результати, розробити алгоритм і програму на алгоритмічній мові, виконати переклад програми з алгоритмічної на машинну мову за допомогою транслятора, виконати машинну програму на ЕОМ та отримати результати.

Обчислювальний процес називається лінійним, якщо усі його операції виконуються послідовно в порядку їхнього запису (рис. 2.1).

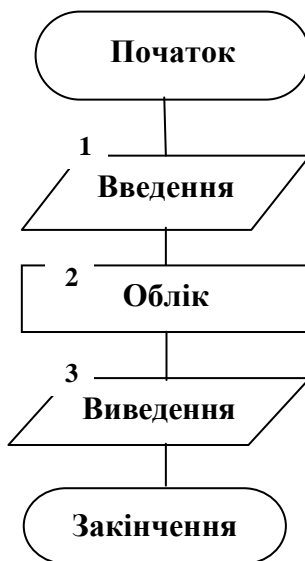


Рисунок 2.1 – Алгоритм лінійного обчислювального процесу

Приклади:

1. Визначити ціну товару в іншій валюті, якщо відомий її курс.
2. Визначити залишок товару на складі у кінці дня, якщо відома його кількість на початку дня, а також надходження і витрати протягом дня.

Хід роботи

Для того, щоб внести зміни в інтерфейс користувача, необхідно ви-кликати діалогове вікно **Customize**, показане нижче, клацнувши правою кнопкою миші (ПКМ) в області панелі інструментів та вибравши у випадаючому меню пункт **Customize** (рис. 2.2).

Використовуйте вкладку **Toolbars**, щоб створювати, видаляти, переіменовувати, відновлювати панелі інструментів, щоб додавати їх до інтерфейсу програми.

ПРИМІТКА. Вкладка **Commands** відображує команди та меню, які ви можете перетягнути з діалогового вікна **Customize** на панель інструментів. Вкладка **Options** містить параметри безпеки та інші установчі параметри.

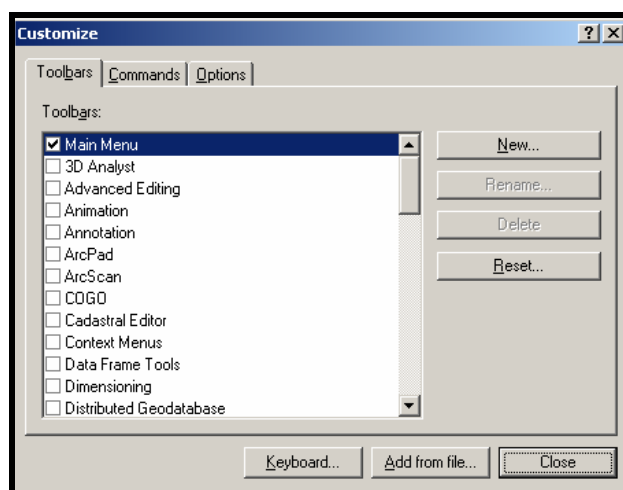


Рисунок 2.2 – Діалогове вікно зміни інтерфейсу Customize

Вправа 1

Постановка задачі

Припустимо що Ви – програміст GIS для відділу планування Манхеттена, штату Канзас. Коли люди приходять в офіс відділу планування для отримання карти ділянок, співробітник відділу використовує ArcMap, щоб створити необхідні карти.

Для скорочення витрат робочого часу співробітників для даного виду роботи, ви створюєте додаток «Огляд ділянок». Користувачі додатку «Огляд ділянок» повинні знайти ділянки, виділити їх номерами і характеристиками; змінити розмір вікна на них; панорамувати, щоб центрувати ці ділянки та друкувати карту. Користувачу також необхідно змінювати масштаб, щоб бачити відносне розташування ділянки на карі міста. В програмі ArcMap, ви створите панель інструментів з командами саме для цих дій.

Запустіть програму **ArcMap** та відкрийте файл **ex02a.mxd** у папці **C:\ArcObjects\Chapter02**. Збережіть документ у своїй папці і лише після цього можна вносити до нього зміни. Карта містить шари для міської межі, ділянок і вулиць.

У меню **Tools** виберіть **Customize**, перейдіть на вкладку **Toolbars**.

Список панелей інструментів буде змінюватися залежно від того, яке розширення **ArcGIS** Ви завантажили на свій комп'ютер.

На вкладці **Toolbars** натисніть на кнопку **New**, для того щоб викликати діалогове вікно для створення нової панелі інструментів. Нова панель інструментів буде збережена в документі карти **ex02a.mxd**.

Змініть назву панелі інструментів на **Parcel Viewer** («Огляд ділянок») та виберіть з випадаючого списку **Save in** назву файлу, в якому ви знаходитесь – **ex02a.mxd** (рис. 2.3). Натисніть кнопку **OK**.

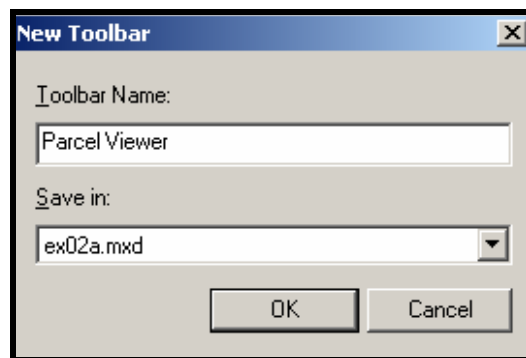


Рисунок 2.3 –Вікно створення нової панелі інструментів

Панель **Parcel Viewer** буде додано до списку панелей інструментів в діалоговому вікні **Customize**. Панель інструментів, до якої не додано жодної команди буде змінювати свої розміри в залежності від її наповнення.

Тепер необхідно перемістити команду **Find** з панелі інструментів **Tools** на створену нами панель інструментів **Parcel Viewer**. Цей інструмент співробітники будуть використовувати для пошуку необхідної ділянки по її номеру – ID.

ПРИМІТКА. Коли діалогове вікно **Customize** відкрите, просто перетягніть інструмент **Find** на панель інструментів **Parcel Viewer**. (Якщо панель інструментів **Tools** не відображається, Ви можете викликати її за допомогою діалогового вікна **Customize**).

У діалоговому вікні **Customize** перейдіть на вкладку **Commands**. Це вікно містить два списки: **Categories** (категорії) та **Commands** (команди).

У списку **Categories** (категорії), переміщаючись вниз виберіть категорію **Pan/Zoom** (рис. 2.4).

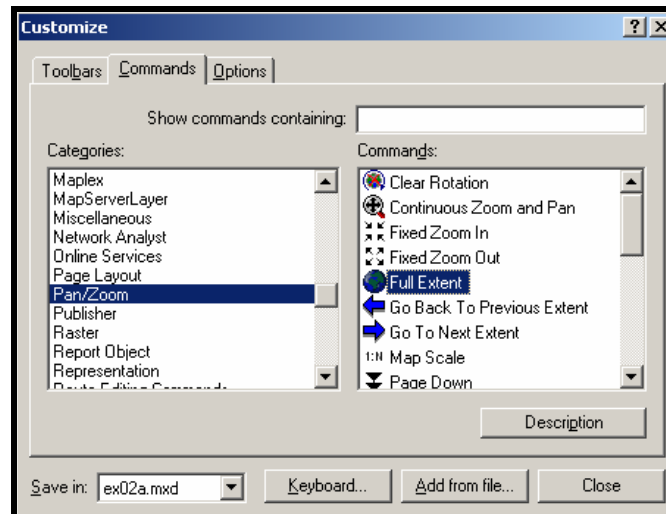


Рисунок 2.4 – Вибір команди для додавання на панель інструментів

Із списку **Commands** (команди) перемістіть команду **Full Extent** на панель інструментів **Parcel Viewer**, перетягнувши її лівою кнопкою миші (ЛКМ).

Додайте команди **Zoom In** та **Pan** з цієї ж категорії на панель інструментів **Parcel Viewer**, а потім з категорії **View** додайте команди **Data View** та **Layout View**.

Оскільки деякі користувачі можуть бути не знайомі із стандартними зображеннями на кнопках **ArcMap**, далі ви навчитесь замість зображень використовувати текст на прикладі кнопок **Data View** та **Layout View**.

На панелі інструментів **Parcel Viewer**, правою кнопкою миші клацніть по кнопці **Data View** та виберіть **Text only** (тільки текст). Зображення на кнопці зникає і на панелі залишається тільки текст

Повторіть свої дії для кнопки **Layout View**.

Тепер на створеній вами панелі інструментів відображаються дві команди з іменами замість зображень. Щоб назви кнопок не зливалися необхідно додати роздільник.

На створеній панелі інструментів **Parcel Viewer** клацніть правою кнопкою миші на кнопці **Data View** і виберіть у випадаючому меню команду **Begin a Group** (почати групу). Зробіть те ж для команди **Layout View**.

У діалоговому вікні **Customize** перейдіть на вкладку **Commands** та у

списку **Categories** виберіть **File**. Із списку **Commands** виберіть команду **Print** та перетягніть її ЛКМ на панель інструментів **Parcel Viewer**. Створіть зліва від цієї команди роздільник. Ваша панель інструментів повинна тепер виглядати як на рисунку 2.5.

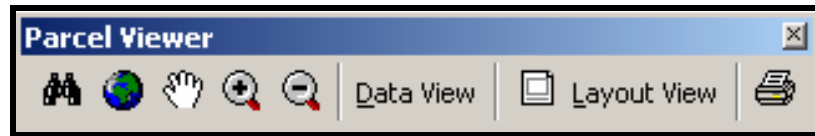


Рисунок 2.5 – Створена панель інструментів **Parcel Viewer**

Панель інструментів **Parcel View** тепер містить усі команди, які Вам потрібні щоб знайти необхідний пакет і надрукувати карту. Всі інші панелі інструментів можна вимкнути. В діалоговому вікні **Customize** відключіть всі панелі інструментів окрім **Main Menu** та **Parcel View**. І хоча вам основне меню не потрібне ArcMap вимагає, щоб воно було присутнім.

Закрийте діалогове вікно **Customize**. Розташуйте створену панель інструментів **Parcel View** нижче **Main Menu**.

Протестуйте панель інструментів **Parcel View**. На панелі інструментів **Parcel View**, натисніть кнопку **Find**.

- В області випадаючого списку **Find** вікна **Find**, наберіть код ділянки 71400.
- Рядок знайденої ділянки з'явиться у списку знайдених об'єктів (рис. 2.6).

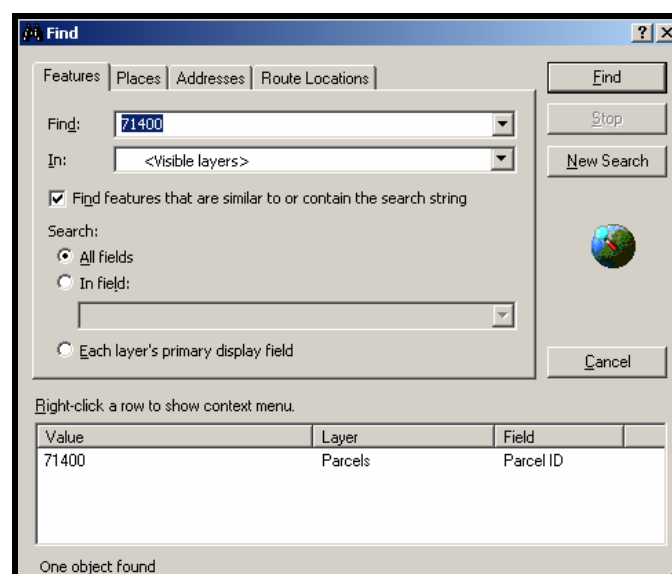


Рисунок 2.6 – Пошук ділянки за допомогою вікна Find

У списку знайдених об'єктів правою кнопкою миші клацніть на знайденій ділянці та виберіть команду **Flash**. В результаті ділянка 71400 виділиться на карті зеленим кольором.

У вікні **Find**, клацніть правою кнопкою миші на знайденій ділянці та виберіть команду **Zoom to**, а потім у випадаючому меню виберіть команду **Select**.

Закрийте вікно **Find**.

На панелі інструментів **Parcel View**, натисніть на кнопку команди **Layout View**, щоб побачити який вигляд має карта.

Карта ділянки відобразилася і готова до друку (рис. 2.7).

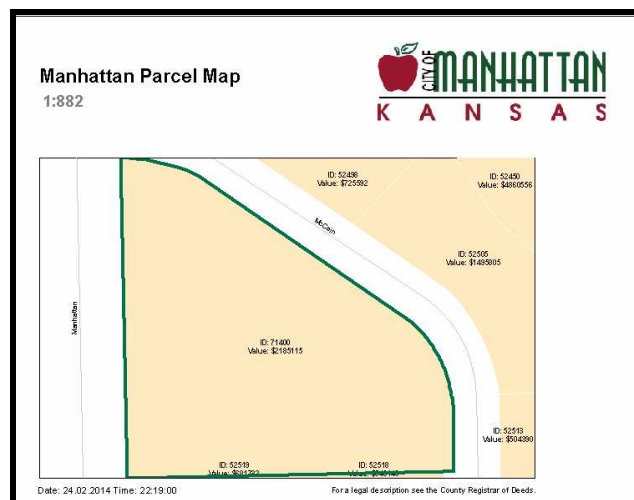


Рисунок 2.7 – Карта знайденої ділянки

ПРИМІТКА. Під час виконання вправи, Ви вимкнули усі панелі інструментів окрім **Main Menu** та **Parcel View**. Ви можете включити їх знов у будь-який час, але вони вам не знадобляться в наступній вправі.

Вправа 2

Постановка задачі

Припустимо що співробітники відділу планування використовують ваш додаток «Огляд ділянок» для перегляду і друку карт земельних ділянок. Тепер їм необхідна допомога у знаходженні необхідних ділянок і друку карт. Ви створите користувацьку панель, яка підкаже користувачу як себе вести, якщо щось незрозуміло.

В цій вправі Ви навчитеся створювати кнопку панель. Ви напишете програмний код, який буде пов'язаний з кнопкою на панелі, натиснувши на яку користувач отримає необхідну інформацію.

Запустіть ArcMap і відкрийте проект **ex02b.mxd** за адресою **C:\ArcObjects\Chapter02**.

Коли проект відкриється, ви побачите земельні ділянки Манхеттену та панель інструментів **Parcel Viewer**.

В ArcMap відкрийте діалогове вікно **Customize** та перейдіть на вкладку **Commands**. В списку категорій виберіть **UIControls**. Перевірте, що ім'я вашого файлу виbrane у випадяючому списку **Save in**. Список команд зараз порожній. Як тільки ви створите команду користувача, назва команди з'явиться у списку **Commands**.

Клацніть на кнопці **New UIControl**.

За допомогою радіо-кнопок виберіть **UIButtonControl** і клацніть на кнопку **Create** (створити) (рис. 2.8).

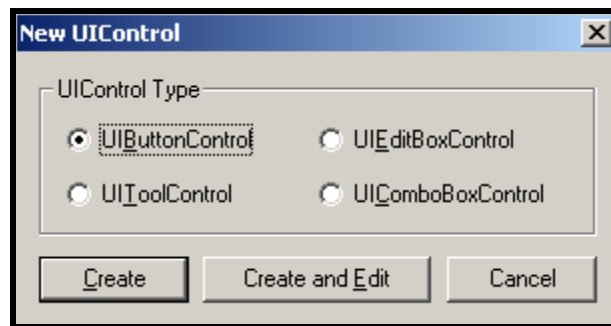


Рисунок 2.8 – Вікно створення нової кнопки

Ви бачите нову кнопку, **Project.UIButtonControl**, в списку команд. Ім'я кнопки має префікс «**Project**», тому що ви вибрали ім'я файлу **ex02b.mxd** в списку **Save in**.

ПРИМІТКА. Якщо у випадяючому списку **Save in** ви вибрали шаблон **Normal.mxt** ви б отримали назву кнопкової панелі з префіксом «**Normal**». Оскільки, ви зберегли кнопку панелі в проекті (файлі) **ex02b.mxd**, то ви і ваші користувачі повинні відкрити цей проект, щоб скористатися функціями створеної панелі інструментів. А **Normal.mxt** – це файл, який ArcMap прочитує кожного разу при запуску. Будь-які збережені вами панелі в шаблон **Normal.mxt** з'являються завжди у бідь-якому відкритому файлі.

В списку команд (**Commands**) діалогового вікна **Customize** натисніть на назву кнопки **Project.UIButtonControl1** і змініть ім'я на **Project.Help** (рис. 2.9).

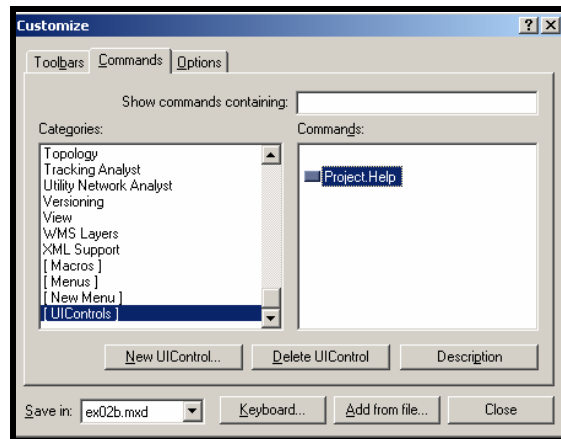


Рисунок 2.9 – Перейменування нової кнопки

Перетягніть створену кнопку **Project.Help** на панель інструментів **Parcel Viewer**.

Тепер Ви зміните зображення кнопок так, що будуть показані тільки їхні назви. Не закриваючи діалогове вікно **Customize** клацніть правою кнопкою миші на панелі і виберіть з випадаючого меню команди **Text Only**, а потім **Begin a Group**, щоб додати розділову лінію.

Тепер Ви напишете код VBA, щоб ваша кнопка **Help** працювала. Клацніть правою кнопкою миші на кнопці **Help** і виберіть **View Source**.

Відкриється вікно редактора VBA. Воно складається з інших вікон, таких як **Project**, **Properties** і вікна для запису коду, яке називається **ThisDocument (Code)**. Виходячи з настройок VBA, це вікно може мати різні розміри, а його розташування регулюється перетягуванням ЛКМ.

Процедури обробки подій для **UIControls** проекту збережені в кодовому вікні **ThisDocument**. Після клацання ЛКМ на елементі – кнопці **UIButton** до вікна з кодом додається автоматично процедура обробки події клацання на кнопці, тобто **Click()** (рис 2.10).

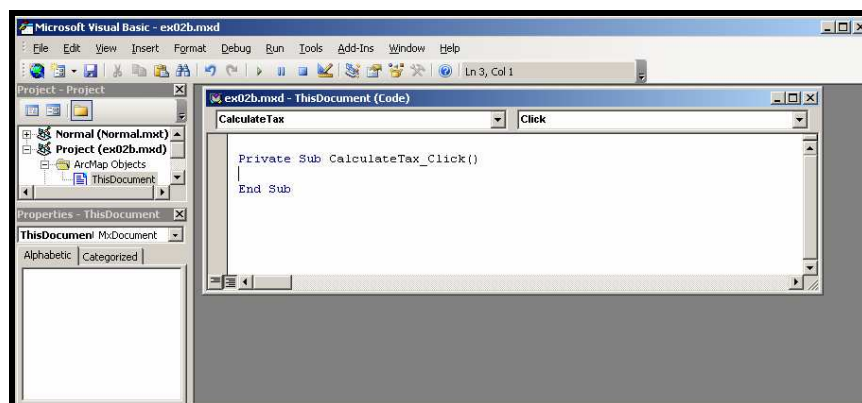



Рисунок 2.10 – Вигляд вікна редактора VBA

Між рядками початку і закінчення процедури, наберіть наступний код:

msgbox "Для отримання допомоги подзвоніть у відділ програмістів 123-45-67"

Протестуйте отриману вами процедуру. Для цього на стандартній панелі інструментів, клацніть кнопку **Run Sub**  (або клавішу клавіатури **F5**).

Строчка коду працює, ArcMap переходить на передній план і з'являється повідомлення про допомогу (рис. 2.11).

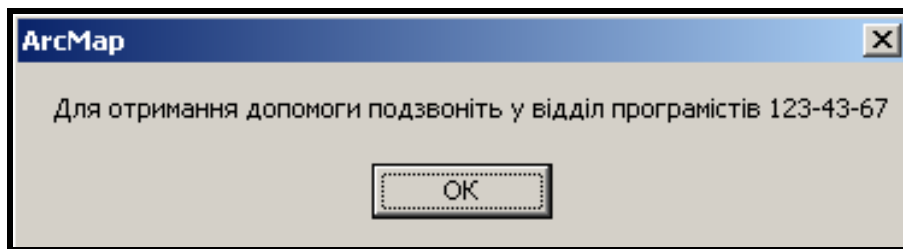


Рисунок 2.11 – Результат роботи процедури з msgbox

Один із способів тестування процедури не залишаючи редактор Visual Basic – це виділити процедуру та клацнути ЛКМ кнопку **Run** (запустити). Але користувачеві ArcMap незручно кожен раз заходити до редактора Visual Basic, тому необхідно спробувати викликати повідомлення по іншому (так як буде це робити користувач).

Клацніть по кнопці **OK** у вікні повідомлення. Закрийте редактор Visual Basic. Клацніть ЛКМ по кнопці **Help** на панелі інструментів **Parcel Viewer**. Повинно відобразитися те ж повідомлення.

В цій вправі Ви створили кнопку користувача, помістили її на панель інструментів і написали код VBA обробки події – клацання ЛКМ по кнопці.

Якщо Ви бажаєте зберегти вашу роботу, у меню **File** в ArcMap виберіть команду **Save As**. Вкажіть місцезнаходження та ім'я файлу.

Вправа 3

Постановка задачі

Припустимо що співробітники відділу використовують ваш додаток «Огляд ділянок» для знаходження ділянки і роздруку її карти. Істотним розширенням функціональності сприяло введення в додаток функції розрахунку податку на ділянку, оскільки до цих пір цей розрахунок виконувався за допомогою калькулятора. Таким чином, доцільно створити кнопку для розрахунку податку і написати відповідний код обробки події – клацання ЛКМ по ній.

Запустіть програму ArcMap та відкрийте вправу ex02c.mxd або продовжуйте працювати в тому ж документі, який створено після виконання попередньої вправи.

Створіть нову кнопку – **UIControls** типу **UIButtonControl** і назвіть її **CalculateTax** (рис. 2.12).

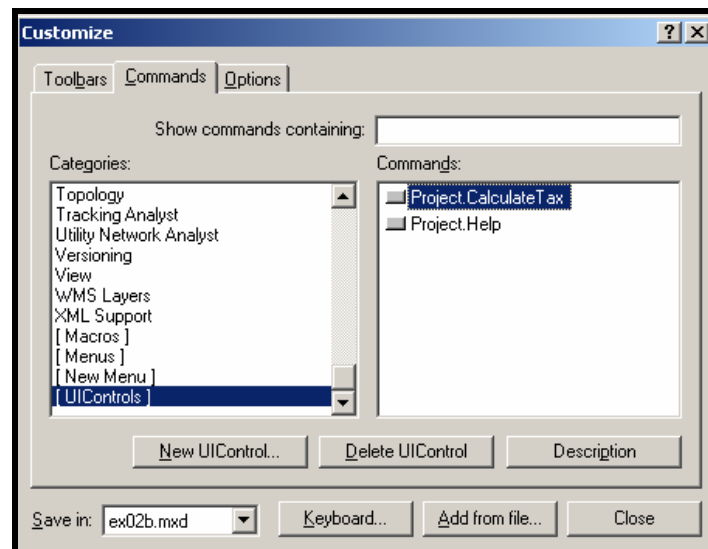


Рисунок 2.12 – Створення нової кнопки

Перетягніть створену кнопку на панель **Parcel Viewer** праворуч від кнопки **Data View**.

Змініть малюнок створеної кнопки на малюнок долара, для цього відкрийте контекстне меню клацанням ПКМ по створеній кнопці. У випадяючому меню, що з'явилося, виберіть команду **Change Button Image**. В діалозі, що відкрився, виберіть файл **Dollar.bmp** за адресою **C:\ArcObjects\Data** (рис. 2.13).

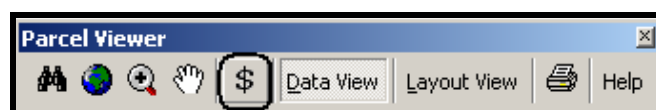


Рисунок 2.13 – Зображення замість назви на кнопці

Перейдіть в редактора коду VBA за допомогою контекстного меню на кнопці та команди **View Source**.

Між командами початку і кінця процедури введіть наступний код для оголошення необхідних змінних розрахунку:

```
Dim curParcelvalue As Currency  
Dim curTaxValue As Currency  
Dim datToday As Date
```

Введіть наступний код (символ підкреслення використовується для переходу на новий рядок), або увесь вираз напишіть в один рядок:

```
curParcelValue = InputBox (_  
    "Введіть вартість ділянки", _  
    "Parcel Viewer", 100000)
```

Введіть наступний код для обчислення податку з ділянки:

```
curTaxValue = (curParcelValue * 0.02) + 8.55 + 11
```

Ділянки під забудову оподатковуються 2% від їхньої вартості плюс 8,55\$ страхового внеску та 11\$ нотаріального збору .

Введіть наступний код для присвоєння значення змінній дати.

```
datToday = Now
```

«**Now**» це вбудована функція Visual Basic.

Введіть наступний код для виведення повідомлення на екран про розмір податкової ставки на сьогоднішній день:

```
MsgBox "Податок на ділянку: $" & curTaxValue, _  
    vbInformation, datToday
```

Після виконання завдань у вас повинен бути сформований код, який наведено нижче (рис. 2.14):

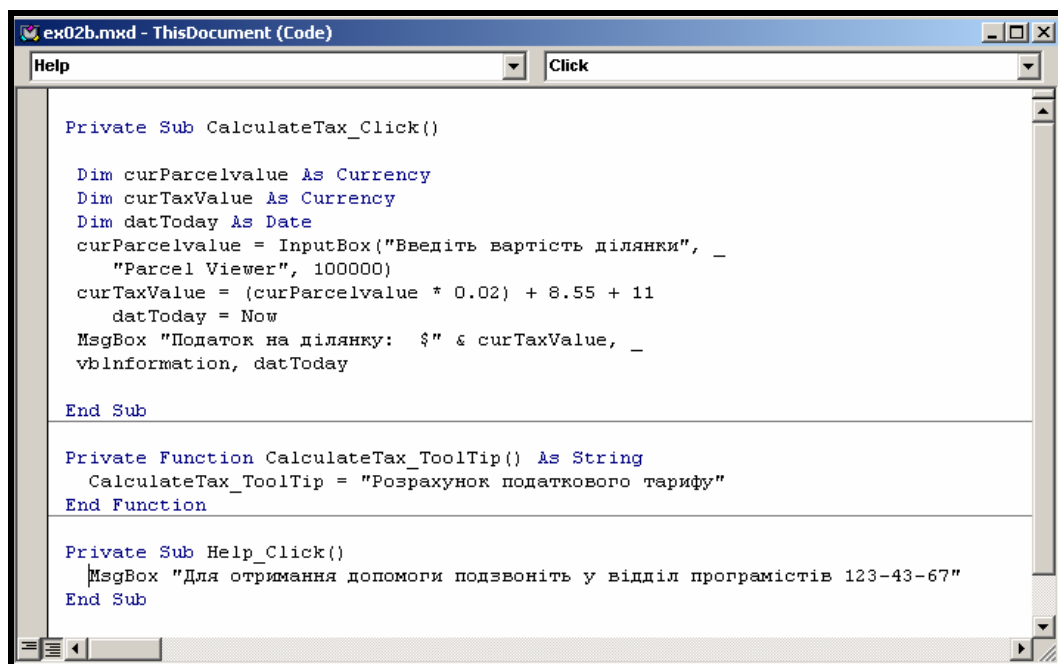


Рисунок 2.14 – Вікно з кодом

Закрийте редактор **Visual Basic**. Протестуйте кнопку **CalculateTax**, дотримуючись інструкцій, які наведені нижче.

На панелі інструментів **Parcel Viewer**, клацніть по кнопці **Calculate Tax (\$)**. Змініть число 10000 на 15000. У вікні з'явиться результат розрахунку (рис. 2.15). Клацніть по кнопці **OK**.

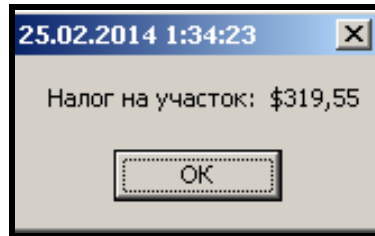


Рисунок 2.15 – Вікно з результатом розрахунку

Ви отримали розрахований податок. Отже, виконуючи вправи Ви створили два власні елементи управління (кнопки) типу **UIButtonControl** та коди обробників події **Click**. Далі ми створимо код для обробки події **Впливаюча підказка – ToolTip**.

ПРИМІТКА. Виклик обробника події **ToolTip** відбувається тоді, коли користувач наводить ЛКМ на команду, але не клаціє по ній.

Нижче на малюнку показана ситуація, коли користувач наводить ЛКМ на команду **Zoom In** з'являється впливаюча підказка з ім'ям команди (рис. 2.16).

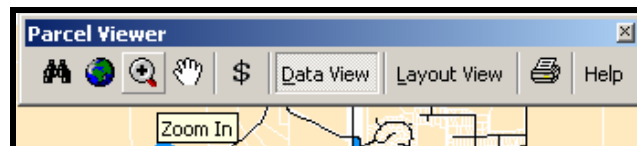


Рисунок 2.16 – Впливаюча підказка для команди **Zoom In**

Клацніть правою кнопкою миші по кнопці **CalculateTax** і виберіть команду **View Source** (вікно **Customize** повинно бути відкрито).

Ви побачите процедуру події – **Click** і код, який ви вже додали. Щоб додати інший обробник подій для кнопки **CalculateTax**, ви повинні вибрати його із списку події (він знаходиться праворуч) (рис. 2.17).

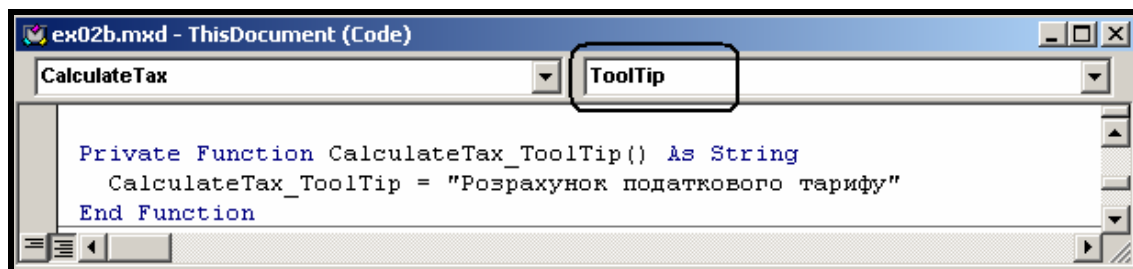


Рисунок 2.17 – Вибір події зі списку

В списку вибору типу події клацніть стрілку вниз і з випадуючого списку виберіть **ToolTip**. Між рядками початку та закінчення процедури введіть відповідний код:

CalculateTax_ToolTip = "Расчет налогового тарифа"

Закрийте редактор Visual Basic. Підведіть ЛКМ до кнопки **CalculateTax**, щоб бачити випадуючу підказку з описом інструменту. Збережіть Вашу роботу (файл ArcMap).

Завдання для самостійного виконання

1. Створити власну панель інструментів для роботи з картою світу.
2. Заздалегідь створити необхідний документ *.mxd, включивши до нього шари City, Cntry00. Панель інструментів повинна носити назву: «ПІБ. Варіант № __», наприклад: «Іванов І.І. Варіант № 3».
3. Залежно від Вашого варіанту у Вашу панель інструментів повинні входити інструменти перераховані у таблиці 2.1 (інструменти задані номерами, розшифровка приводиться в табл. 2.2):

Таблиця 2.1 – Номери інструментів для створення панелі користувача

№ варіанту	Перелік номерів інструментів					
1	2	3	4	5	6	8
2	3	5	8	9	12	13
3	3	4	7	8	9	10
4	3	4	5	7	8	11
5	3	4	7	9	11	14
6	3	4	5	9	11	12
7	1	2	5	9	12	14
8	2	3	5	8	11	12
9	3	5	7	9	12	15
10	2	4	6	10	11	13
11	2	3	5	9	11	14
12	1	3	5	9	12	15
13	1	3	5	8	10	12
14	3	5	7	8	11	14
15	2	3	6	9	11	12

Таблиця 2.2 – Розшифровка номерів інструментів

№ інструменту	Категорія інструменту	Команда (назва інструменту)
1	Selection	Identify
2	Selection	Measure
3	Selection	Find
4	Selection	Select Feature
5	Selection	Select All
6	Selection	Switch Selection
7	Pan/Zoom	Pan
8	Pan/Zoom	Full Extent
9	Pan/Zoom	Zoom In
10	Pan/Zoom	Zoom Out
11	Label	Label
12	Tools	Visual Basic Editor
13	Tools	ArcCatalog
14	View	Data View
15	View	Layout View

4. Додайте до створеної панелі інструментів два роздільники (початок нової групи інструментів) у тому місці, де починаються інструменти з іншої категорії.
5. Замініть зображення одного з інструментів на назву команди.
6. * До створеної панелі інструментів додайте власну кнопку, яка буде виводити довільне повідомлення.
7. Як заголовок для вікна повідомлення використайте Ваше прізвище (третій параметр в команді **MsgBox**). Як спливаючу підказку використайте довільний текст. Зображення кнопки замінити текстом «Повідомлення».
8. Додайте ще одну кнопку, яка буде виконувати переведення введеного числа (за допомогою **InputBox**) з одних одиниць виміру в інші.

Одиниці виміру по варіантах наведені в таблиці 2.3. У якості спливаючої підказки використовувати текст «Переведення з __ в __». Зображення кнопки замінити відповідним малюнком.

Таблиця 2.3 – Одиниці переведення по варіантах

№ варіанту	Вихідні одиниці виміру	Кінцеві одиниці виміру
1	мілі	кілометри
2	кілометри	мілі
3	метри	фути
4	фути	метри
5	дюйми	сантиметри
6	сантиметри	дюйми
7	дюйми	метри
8	метри	дюйми
9	градуси	радіани
10	радіани	градуси
11	гривня	долар
12	долар	гривня
13	гривня	євро
14	євро	гривня
15	градуси Цельсія	градуси Фаренгейта

Контрольні питання

1. З яких вкладок складається вікно настройки інтерфейсу **Customize**?
2. Які параметри можуть бути встановлені на вкладці **Options** вікна настройки інтерфейсу?
3. Які варіанти збереження власних настройок інтерфейсу Ви знаєте?
4. Які пункти з'являються у випадаючому меню при настройці стандартних кнопок графічного інтерфейсу?
5. Які пункти з'являються у випадаючому меню під час настройки створених вами **UIButtonControl** кнопок графічного інтерфейсу?
6. Назвіть типи елементів доступних для створення власних **UIControl**.
7. Ознайомтеся з вікнами редактора VBA, використовуючи для їхнього включення і виключення пункти стандартного меню View. Перерахуйте ці вікна та вкажіть їхнє призначення.
8. Ознайомтеся з пунктами меню **Run**. Перерахуйте різноманітні способи запуску ваших процедур.
9. Які види подій можуть відбуватися із створеними Вами кнопками графічного інтерфейсу?
10. Коли відбувається подія **ToolTip Click**?

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений на персональному комп'ютері (ПК), роздрукований на аркушах формату А4. Звіт з лабораторної роботи повинен містити:

Титульний аркуш.

Оформлення кожної вправи:

- Постановка задачі.
- Опис розв'язання.
- Необхідний лістинг (код програми), скриншоти і блок-схеми алгоритмів (дивись додаток А).

Індивідуальне завдання виконується відповідно номера в журналі групи.

Оформлення кожного завдання:

- Постановка задачі.
- Опис розв'язання.
- Необхідний лістинг (код програми), скриншоти і блок-схеми алгоритмів.
- Відповіді на контрольні питання.

До звіту необхідно додати файл, виконаний у програмному забезпеченні ArcMap з виконаною роботою.

ЛАБОРАТОРНА РОБОТА № 3 НАЛАШТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

Мета: виробити уміння та навички по створенню інтерфейсу користувача за допомогою вбудованої в програму ArcMap мови програмування VBA.

Призначення: виконавши роботу, Ви навчитеся створювати форми користувача для вирішення різноманітних завдань.

Підготовка до роботи

1. Яким чином створювати елементи управління на формі користувача?
2. Яким чином настроїти властивості елементів управління?
3. Призначення функції **MsgBox**. Які аргументи входять до неї? Та їх призначення?

4. Призначення функції **InputBox**. Які аргументи входять до неї? Та їх призначення?
5. В чому полягає відмінність функції **MsgBox** від **InputBox**?

Ключові слова

Інтерфейс, форма користувача, функція, аргумент функції, елемент управління, властивості елемента управління.

Теоретичні відомості

Інтерфейс – це зовнішня оболонка додатку разом з програмами управління доступом та іншими прихованими від користувача механізмами управління, яка надає можливість працювати з документами, даними і іншою інформацією, яка зберігається в комп'ютері. Головна ціль будь-якого додатку – забезпечити максимальну зручність і ефективність роботи з інформацією: базами геоданих, графікою або зображення. Тому інтерфейс є найважливішою частиною будь-якого додатку.

Добре розроблений інтерфейс гарантує зручність роботи користувача з додатком. Проектування інтерфейсу – процес циклічний. На цьому етапі розробки додатку бажано частіше спілкуватися з користувачами і замовниками додатку для розробки найприйнятніших по ефективності та зручності і зовнішнього вигляду інтерфейсного рішення.

Під час роботи в ArcMap, як і в більшості інших додатків, доводиться стикатися з такими елементами інтерфейсу, як діалогові вікна. Діалогові вікна використовуються для отримання інформації для введення і виведення повідомлень та даних. Одним з об'єктів візуалізації VBA є Userform, який призначений для користувача. Призначені для користувача форми – це діалогові вікна інтерфейсу процедур VBA. З допомогою діалогових вікон користувач може ефективно передавати дані в процедури і одержувати результати їхньої роботи. Призначені для користувача форми дають можливість створювати діалогові вікна в додатках, що розробляються, і розміщувати у вікнах елементи управління. В VBA є дві функції – **Msgbox** та **Inputbox**, які дозволяють відображати прості діалогові вікна, не створюючи форму користувача. Ці вікна можна видозмінювати, використовуючи керовані ними параметри, але вони не мають тих широких і ефективних можливостей та опцій, які надають форми користувача.

VBA пропонує великі можливості, які можна використовувати при створенні призначених для користувача діалогових вікон та програмування елементів управління.

Хід роботи

Вправа 1

Постановка задачі

Необхідно створити інтерфейс у вигляді форми користувача для розрахунку ставки оподаткування в залежності від зони розташування земельної ділянки та її вартості.

Запустіть **ArcMap**. Виберіть в меню **Tools** пункт **Macros**, а в ньому **VisualBasicEditor**. У вікні редактора VBA необхідно створити нову форму. Для цього в меню **Insert** виберіть команду **UserForm**. Потім у вікні **Property** необхідно змінити властивості форми наступним чином (табл. 3.1).

З палітри доступних елементів управління **ToolBox** перетягуємо елемент **Image** та налаштовуємо властивості цього елемента за допомогою вікна властивостей **Properties** (табл. 3.2).

Далі додамо текстове поле, поле зі списком та мітку. З палітри **ToolBox** перетягуємо елемент форми – **TextBox**, **ComboBox** та **Label**, і розміщуємо їх в будьякому місці на формі. Вибираємо кожний елемент по черзі та встановлюємо наступні властивості для кожного окремо (табл. 3.3).

Таблиця 3.1 – Значення властивостей форми

Властивість	Значення	Опис властивості
Name	frmTax	Внутрішнє ім'я об'єкту
BackColor	Виберіть білий колір з випадуючого списку	Колір фону
Caption	Tax Calculator	Заголовок, який бачить користувач
Height	200	Висота об'єкту у пікселях
Width	300	Ширина об'єкту у пікселях

Таблиця 3.2 – Властивості елементу **Image**

Властивість	Нове значення	Опис властивості
1	2	3
Name	imgLogo	Внутрішнє ім'я об'єкту
BackColor	Виберіть білий колір з випадуючого списку	Колір фону

Продовження таблиці 3.2

1	2	3
BorderColor	Виберіть білий колір з випадального списку	Колір границі об'єкту
Height	60	Висота об'єкту у пікселях
Left	18	Ліва координата об'єкту
Top	6	Верхня координата об'єкту
Width	216	Ширина об'єкту у пікселях
Picture	Виберіть придатний малюнок	Зображення малюнку (для його підключення вкажіть місцезнаходження)
PictureSizeMode	0	Масштабування малюнку під час зміни розміру рамки малюнку

Таблиця 3.3 – Властивості елементів форми

Властивість	Нове значення	Опис властивості
TextBox (текстове поле)		
Name	txtParcelValue	Внутрішнє ім'я об'єкту
Height	18	Висота об'єкту у пікселях
Left	96	Ліва координата об'єкту
Top	200	Верхня координата об'єкту
Width	150	Ширина об'єкту у пікселях
ComboBox (поле з випадальним списком)		
Name	cboZoning	Внутрішнє ім'я об'єкту
Height	18	Висота об'єкту у пікселях
Left	96	Ліва координата об'єкту
Top	102	Верхня координата об'єкту
Width	150	Ширина об'єкту у пікселях
Label (мітка)		
Name	lblTaxAmount	Внутрішнє ім'я об'єкту
Height	18	Висота об'єкту у пікселях
Left	96	Ліва координата об'єкту
Top	126	Верхня координата об'єкту
Width	72	Ширина об'єкту у пікселях
Caption	Видалити текст з мітки, щоб не було ніякого напису	Напис, який видно на мітці
Font	FontStyle=Bold	Параметри шрифту

З палітри **ToolBox** додаємо ще три мітки і розміщуємо їх як зображено на рисунку 3.1.

У вікні **Properties**, встановлюємо властивості для трьох міток наступним чином (табл. 3.4).

Таблиця 3.4 – Властивості міток форми

Мітка	Властивість Name	Властивість Caption
Label1	lblValue	Введіть віртність ділянки
Label2	lblZoning	Вкажіть зону
Label3	lblTax	Розрахована ставка податку

В результаті одержуємо такий зовнішній вигляд форми (рис. 3.2).

ПРИМІТКА. Виділяємо усі три мітки (клацаємо по черзі ЛКМ утримуючи клавішу **Shift**). Для того щоб вирівняти їх по правому краю у вікні **Properties** обираємо властивість **TextAlign** і з випадаючого списку обираємо значення **3-fmTextAlignRight**.

Додамо на форму кнопку для розрахунку і кнопку, щоб закрити діалогове вікно. Для цього з палітри **Toolbox** ЛКМ перетягуємо на форму дві **CommandButtons** (рис. 3.3).

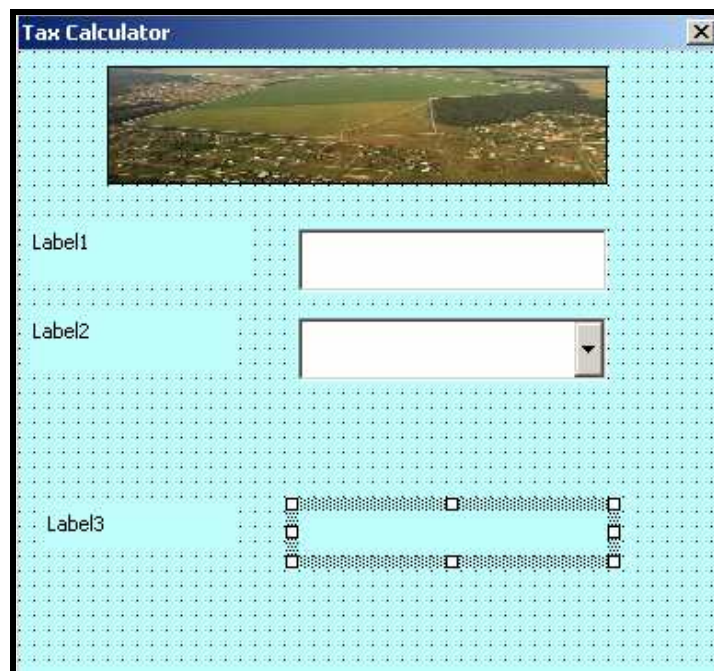


Рисунок 3.1 – Зовнішній вигляд форми **frmTax**

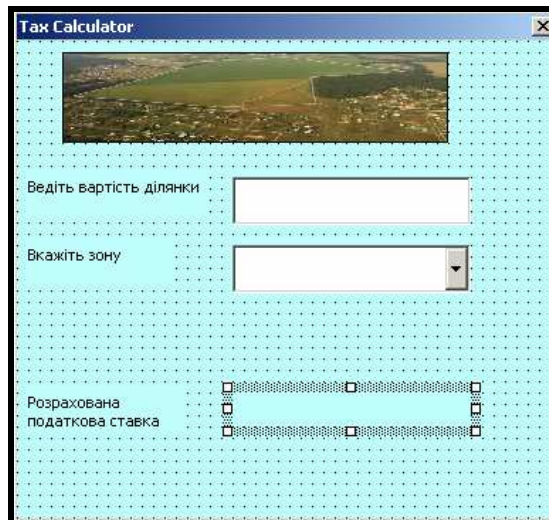


Рисунок 3.2 – Зовнішній вигляд форми **frmTax** після введення властивості **Caption** для міток

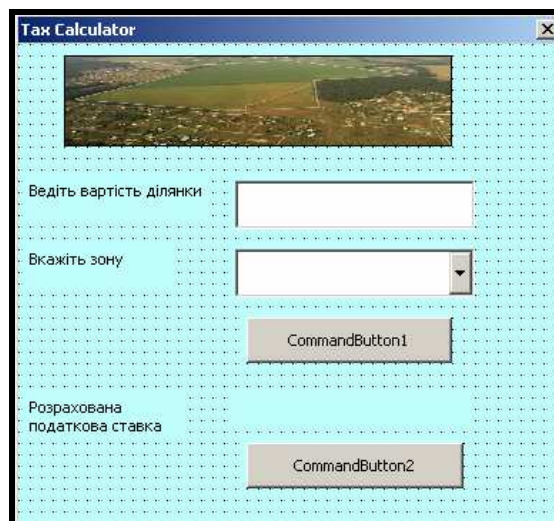


Рисунок 3.3 – Зовнішній вигляд форми «**frmTax**» після додавання кнопок

Для кожної кнопки **CommandButton**, встановлюємо наступні властивості (табл. 3.5).

Таблиця 3.5 – Властивості міток форми

Кнопки	Властивість Name	Властивість Caption
Розрахунок	cmdCalculateTax	Розрахунок податку
Вихід	cmdQuit	Вихід

Для перевірки створеного інтерфейсу форми натискаємо кнопку **Run Sub/User Form**. Форма з'являється як діалогове вікно (рис. 3.4).

Рисунок 3.4 – Остаточний зовнішній вигляд форми «frmTax»

Завдання для самостійного виконання

1. Створіть форму за допомогою якої буде виконуватися переведення введеного числа з одних одиниць вимірювання в інші. Варіанти наведено у таблиці 3.6. Передбачити, щоб одна одиниця виміру переводилася не менше ніж у три результуючі.
2. Для виклику форми створіть кнопку. У якості спливаючої підказки використайте текст «**Переведення з __ в __**». Зображення кнопки замінити відповідним малюнком.

Таблиця 3.6 – Варіанти завдань для створення форми

№ варіанту	Тема перерахунку
1	2
1	Обмін валют: внутрішньоєвропейські
2	Калькулятор мір маси
3	Калькулятор мір енергії
4	Калькулятор мір сили
5	Калькулятор виміру нафти і газу
6	Калькулятор мір площі
7	Калькулятор мір кутів
8	Калькулятор мір тиску

Продовження таблиці 3.6

1	2
9	Калькулятор мір довжини: метричні – англійський
10	Калькулятор мір довжини: метричні – російські
11	Калькулятор мір потужності
12	Обмін валют: міжконтинентальні
13	Калькулятор мір об'єму
14	Калькулятор мір швидкості
15	Тригонометричний калькулятор

3. На формі передбачити наступний набір елементів:
 - елемент для введення початкової одиниці виміру;
 - елемент для вибору результуючих одиниць або функцій обчислення;
 - елемент для представлення результату розрахунку.
4. Крім того, додати на форму дві кнопки:
 - кнопку для активізації обчислення;
 - кнопку для закриття форми.
5. Додати до форми відповідний малюнок.
6. Оформити форму на свій розсуд.
7. Протестувати створений інтерфейс форми (поки що без обробників подій).
8. Підготувати формули для перекладу заданих по варіанту одиниць виміру.

Контрольні питання

1. Назвіть, які властивості елемента управління відповідають його іменам – зовнішньому і внутрішньому?
2. Для якого елемента управління зовнішнє ім'я є основною властивістю, заради якого цей елемент і використовується?
3. Перерахуйте властивості елемента управління, які вказують його місцезнаходження на формі.
4. Запишіть назви усіх доступних на палітрі **ToolBox** елементів управління, скориставшись спливаючими підказками та охарактеризуйте їх призначення.

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений на ПК, роздрукований на аркушах формату А4. Звіт з лабораторної роботи повинен містити (дивись лаб. роботу № 2). До звіту необхідно додати файл, виконаний у програмному забезпеченні ArcMap з виконаною роботою.

ЛАБОРАТОРНА РОБОТА № 4

ПРОГРАМУВАННЯ ОБ'ЄКТІВ ФОРМИ.

РОЗГАЛУЖЕНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ

Мета: виробити уміння і навички розробки алгоритмів та програмного коду розгалужених обчислювальних процесів за допомогою вбудованої в програму ArcMap мови програмування VBA.

Призначення: виконавши роботу, Ви навчитеся створювати обчислювальні процеси з розгалуженням, для вирішення різноманітних завдань.

Підготовка до роботи

1. Чим відрізняється однорядковий оператор **If Then** від багаторядкового?
2. Куди передається управління, якщо умова в операторі **If Then Else** істинна?
3. Куди передається управління, якщо умова в операторі **If Then Else** хибна?
4. Які події використовуються у формі для виконання процедур?

Ключові слова

Форма користувача, функція, обчислювальний процес, процес з розгалуженням, оператор.

Теоретичні відомості

Якщо в результаті виконання програми, процес обчислення може пройти тільки по одній з декількох альтернативних колій, то такий обчислювальний процес називається розгалуженим. Наприклад, якщо студент іногородній, то йому надається гуртожиток, якщо ні, то гуртожиток не надається.

На практиці часто зустрічаються задачі, в яких залежно від первинних умов або проміжних результатів необхідно виконати обчислення по одній або іншій формулі. Такі задачі можна описати за допомогою алгоритмів структури, що розгалужуються. В таких алгоритмах вибір напряму продовження обчислення здійснюється за підсумками перевірки заданої умови. Алгоритм, що розгалужується, містить одне або декілька логічних умов і має декілька гілок обробки. Типова структура алгоритму з розгалуженням наведена на рисунку 4.1. Такі процеси описуються оператором IF (умова).

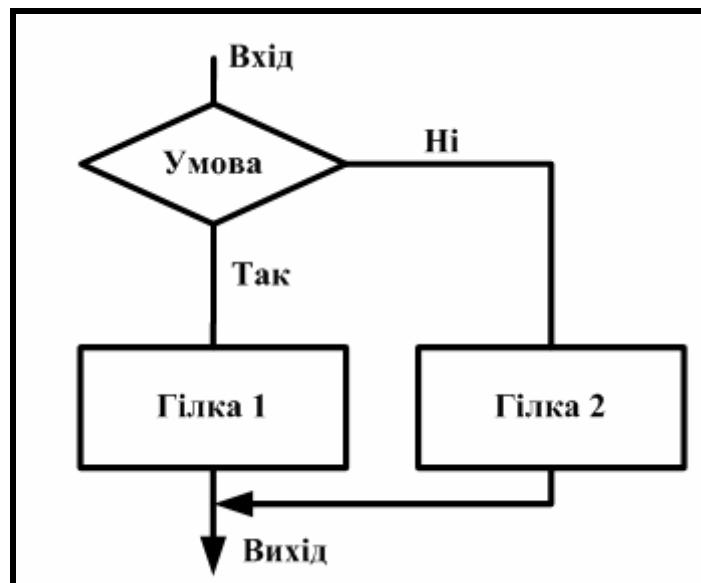


Рисунок 4.1 – Зображення алгоритму з розгалуженням

Умовний оператор – це приклад складного оператора, тобто всередині можуть розташовуватися інші оператори. При застосуванні умовного оператора потрібно пам'ятати наступні правила:

- умовний оператор визначається ключовим словом **IF**;
- після цього ключового слова вказується умова – вираз, при обчисленні якого виходить логічне значення;
- за умовою йде ключове слово **THEN**.

Умова показує, яка група операторів виконується за ключовим словом **THEN**, якщо значення умови – істина, то виконується одна група операторів, якщо помилкова, то виконується інша група операторів.

Кінець розгалуження визначається спеціальним оператором, який складається з двох ключових слів – **END IF**.

Структури операторів дозволяють управляти послідовністю виконання програми. Без операторів управління усі оператори програми будуть виконуватися зліва направо та зверху вниз. Але іноді потрібно багато разів виконувати деякий набір інструкцій автоматично, або вирішувати задачу по-іншому залежно від значення змінних або параметрів, заданих користувачем під час виконання. Для цього служать конструкції управління та цикли.

VBA підтримує такі конструкції умовного оператора:

If . . . Then

If . . . Then . . . Else

Select Case

Конструкція **If . . . Then**

Конструкція **If . . . Then** застосовується, коли необхідно виконати один або групу операторів залежно від деякої умови. Синтаксис цієї конструкції дозволяє задавати її в одному рядку або в декількох рядках програми:

If умова **Then** вираз

If умова **Then**

вираз

End If

Звичайно умова є простим порівнянням, але вона може бути будь-яким виразом зі значенням що обчислюється. Це значення інтерпретується як **False** (брехня), якщо воно нульове, а будь-яке ненульове розглядається як **True** (істина). Якщо умова істинна, то виконуються усі вирази, що стоять після ключового слова **Then**. Для умовного виконання одного оператора можна використовувати як синтаксис для одного рядка, так і синтаксис для декількох рядків (блокову конструкцію).

Наступні два оператори еквівалентні:

If anyDate < Now Then anyDate = Now

If anyDate < Now Then

anyDate = Now

End If

Необхідно зазначити, що синтаксис оператора **If . . . Then** для одного рядка не використовує оператор **End If**. Щоб виконати послідовність операторів, якщо умова істинна, потрібно використовувати блокову конструкцію **If . . . Then . . . End If**.

If anyDate < Now Then

anyDate = Now

Timer.Enabled = False

End If

Якщо умова помилкова, то оператори після ключового слова **Then** не виконуються, а управління передається на наступний рядок (або рядок після оператора **End If** у блоковій конструкції).

Конструкція **If . . . Then . . . Else**

визначає декілька блоків операторів, один з яких буде виконуватися залежно від умови:

If умова 1 **Then**

вираз 1

ElseIf умова 2 **Then**

вираз 2

. . .

Else

вираз-n

End If

При виконанні спочатку перевіряється умова 1. Якщо вона помилкова, VBA перевіряє наступний вираз 2 і т. д., поки не знайде істинної умови. Знайшовши її, VBA виконує відповідний блок операторів і потім передає управління конструкції, наступної за оператором **End If**. В дану конструкцію можна включити блок оператора **Else**, який VBA виконує, якщо не виконана жодна з умов.

Конструкція **If . . . Then . . . ElseIf** насправді всього лише спеціальний випадок конструкції **If . . . Then . . . Else**. Зазначимо, що в даній конструкції може бути будь-яке число блоків **ElseIf**, або навіть жодного. Блок **Else** можна включати незалежно від присутності або, навпаки, відсутності блоків **ElseIf**.

Хід роботи

Вправа 1

Постановка задачі

Необхідно, за допомогою створеного інтерфейсу у попередній роботі, створити програму користувача для розрахунку ставки оподаткування в залежності від зони розташування земельної ділянки і її вартості.

Продовжуємо роботу з файлом, створеним у попередній лабораторній роботі (було створено інтерфейс для форми **frmTax**).

Запустіть **ArcMap**, викличте контекстне меню на кнопці **CalculateTax**, з випадаючого меню оберіть команду **ViewSource**. В редакторі VBA у вікні програмного коду для події **CalculateTax_Click()** створіть код для виклику форми: **frmTax.Show** (рис. 4.2).

```
Private Sub CalculateTax_Click()
    frmTax.Show
End Sub
```

Рисунок 4.2 – Програмний код для події **CalculateTax_Click()**

Викличте форму **frmTax**, використовуючи для цього вікно Проекту (**Projects**). Подвійним натисканням ЛКМ по кнопці **Вихід** (cmdВихід) створіть процедури обробки події, клацання по кнопці **cmdВихіді**. Для цієї події створіть код – **frmTax.Hide**. Потім пропишіть обрамляючи рядки коду (рядки початку та завершення) методу **Initialize** для об'єкта **UserForm**: у списку об'єктів виберіть **UserForm**, у списку методів – **Initialize**.

Додайте між обрамляючими рядками код для внесення у список потрібних позицій випадаючого списку (рис. 4.3):

- **cboZoning.AddItem** «Житлова зона»
- **cboZoning.AddItem** «Комерційна зона»
- **cboZoning.AddItem** «Промислова зона»

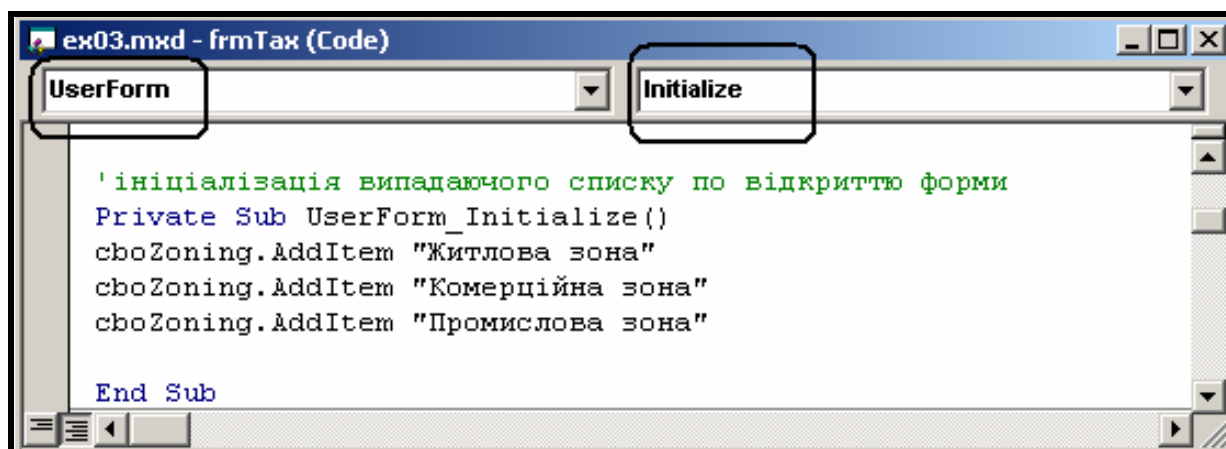


Рисунок 4.3 – Програмний код для створення випадаючого списку

Протестуйте створену форму. Викличте форму натисканням по кнопці **ParcelViewer** на створеній панелі. Перевірте роботу створеного випадаючого списку на формі та закрийте форму натисканням кнопки «Вихід» (рис. 4.4).

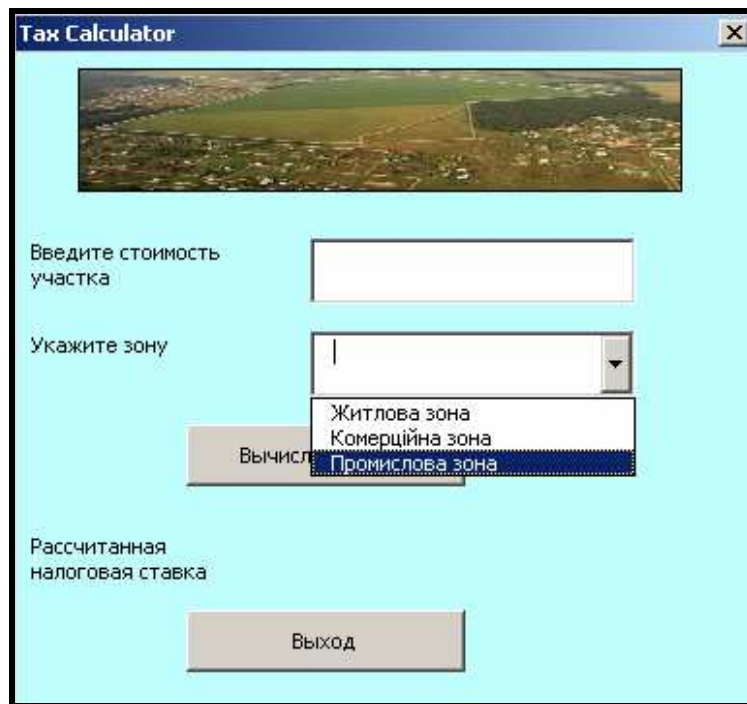


Рисунок 4.4 – Зовнішній вигляд створеної форми

Вправа 2

Постановка задачі

Необхідно створити код для форми **TaxCalculator (frmTax)** для розрахунку значення вартості ділянки, обчислення податку та відображення результату.

Для цього викличте команду **Macros** з меню **Tools** та оберіть команду **VisualBasicEditor**. У вікні редактора VBA проекту повинна відобразитися форма обчислення податку **frmTax**. Подвійним клацанням по кнопці **Облік податку (cmdCalculateTax)** викличте відповідний код обробки події.

Між рядками початку та закінчення процедури, що з'явилися необхідно ввести наступний код (рис. 4.5):

```
Private Sub cmdCalculateTax_Click()
    Dim userValue As Long
    userValue = txtParcelValue.Text
    Dim taxAmount As Long
    taxAmount = (userValue * 0.02) + 8.55 + 11
    lblTaxAmount.Caption = taxAmount
End Sub
```

Рисунок 4.5 – Код для кнопки **cmdCalculateTax**

Тепер необхідно перейти до коду обробника події **cmdQuit_Click()**, щоб додати рядки коду, які будуть очищати елементи управління, розташованого на формі при виході з форми (рис. 4.6).

```
Private Sub cmdQuit_Click()  
  
'обнулення відповідних елементів форми після виходу  
cboZoning.Text = ""  
txtParcelValue.Text = ""  
lblTaxAmount.Caption = ""  
End Sub
```

Рисунок 4.6 – Код для кнопки **cmdCalculateTax**

Протестуйте створену форму: введіть вартість ділянки 250000, після натискання кнопки **Облік податку** повинно з'явитися значення податку 5020.

Вправа 3

Постановка задачі

Ділянки міста належать до зон різних типів, і кожна зона має різну ставку податкового тарифу. Необхідно створити оператор **Case** для обробки вибору користувачем певної зони і розрахунку відповідного податку.

Запустіть ArcMap, запустіть редактора VisualBasic і відкрийте форму для розрахунків **frmTax**.

Необхідно описати перемінну, яка буде відповідати за відмінність у податкових ставках.

ПРИМІТКА. Подвійним клацанням відкрийте обробник події **Click** для кнопки розрахунку податків **cmdCalculateTax**. Введіть необхідні рядки коду для реалізації цього завдання.

Виправіть рядок коду, в якому ведеться обчислення ставки податку (рис. 4.7). Замість 0.02 потрібно ввести перемінну **sngTaxRate**, так щоб вийшло:

$$\text{taxAmount} = (\text{userValue} * \text{sngTaxRate}) + 8.55 + 11$$


```

Private Sub cmdCalculateTax_Click()

    Dim sngTaxRate As Single
    Select Case cboZoning.Value
        Case "Residential"
            sngTaxRate = 0.02
        Case "Commercial"
            sngTaxRate = 0.023
        Case "Industrial"
            sngTaxRate = 0.0275
    End Select

    Dim userValue As Long
    userValue = txtParcelValue.Text

    Dim taxAmount As Long
    taxAmount = userValue * sngTaxRate + 8.55 + 11

    lblTaxAmount.Caption = taxAmount
End Sub

```

Рисунок 4.7 – Код для кнопки cmdCalculateTax

Протестуйте форму. Для цього введіть вартість ділянки 400000, виберіть зону **Промислова** значення податку повинно скласти 11020.

Під час вибору **Комерційна** зона, значення податку повинно складати 9220.

Під час вибору **Житлова** зона значення податку повинно складати 8020.

Вправа 4

Постановка задачі

Необхідно передбачити помилку користувача під час введення у поле чисел текстових записів.

Якщо користувач введе не числове значення в рядок введення вартості ділянки, це спровокує помилку розбіжності типів під час спроби виконати події клацання по кнопці **Вирахувати податок**. Це відбувається тому, що, під час обчислення податку це значення бере участь у множенні. В заданій вправі необхідно написати код, який передбачає цю ситуацію, обробляючи подію **Change** об'єкту **Textbox**. Подія **Change** відбувається завжди, коли ви щось набираєте у рядку введення. Як тільки користувач вносить зміни в рядок, викликається обробник цієї події. Необхідно буде внести в обробник цієї події перевірку того, що користувач набирає у рядку: букви

або цифри. Якщо букви – необхідно заблокувати кнопку **Вирахувати податок**, використовуючи властивість **Enabled**. Коли кнопка заблокована, вона має сірий колір і її не можна натиснути.

Запустіть **ArcMap** і редактор **VisualBasic**, відкрийте форму розрахунку податків **frmTax**. Подвійним клацанням ЛКМ відкрийте обробник події **Change** для елемента редагування **txtParcelValue**.

Всередині рядків початку та завершення процедури (обрамляючих рядків) введіть наступний код (рис. 4.8):

```
'заборона введення тексту у текстове поле txtParcelValue
Private Sub txtParcelValue_Change()

If IsNumeric(txtParcelValue.Text) Then
    cmdCalculateTax.Enabled = True
Else
    cmdCalculateTax.Enabled = False
End If

End Sub
```

Рисунок 4.8 – Код для заборони введення тексту

Протестуйте створену форму, для цього введіть в рядок введення вартості ділянки будь-який текст. Зверніть увагу на те, що кнопка **Обчислення податку** стала заблокованою (сірого кольору) (рис. 4.9).

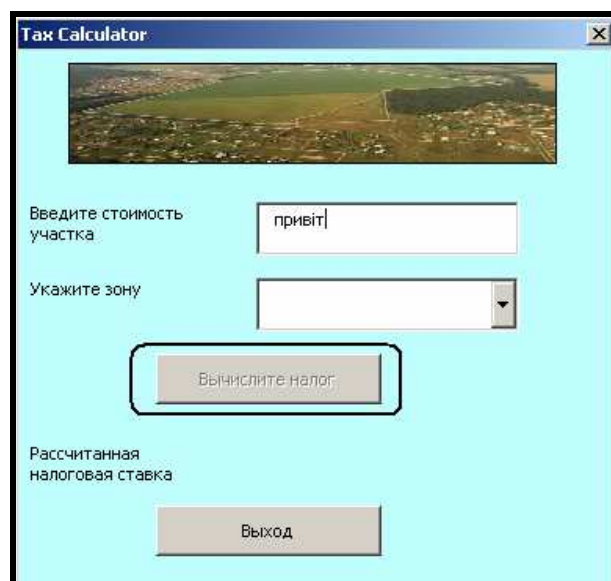


Рисунок 4.9 – Кнопка «Обчислення податку» заблокована

Завдання для самостійного виконання

1. На створеній формі існують тільки три зони земельних ділянок. Змініть код, так щоб обчислення податку провадилося ще й для четвертої зони землекористування.
2. * Поміркуйте над тим, чому під час розрахунку податку з земельної ділянки за допомогою створеного коду, результат не співпадає з розрахунком на звичайному калькуляторі. Які зміни необхідно внести до коду програми, щоб усунути цю невідповідність?
3. Створіть код для форми за допомогою якої буде виконуватися переведення введеного числа з одних одиниць виміру в інші (завдання для самостійного виконання лабораторної роботи № 3).

Контрольні питання

1. Перерахуйте і поясніть призначення подій, які відбуваються, під час відкриття та закриття форми.
2. Навіщо в операторі **Select Case** використовується гілка **Case Else**?
3. Яке призначення та тип результату, який повертається, для функції **IsNumeric()**?
4. Перерахуйте усі функції VBA з префіксом **Is**, використовуючи довідку VBA.
5. Які властивості об'єкту **CommandButton** можуть використовуватися в умовному виразі операторів розгалуження? Перерахуйте їх.

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений за допомогою ПК, роздрукований на аркушах формату А4. Звіт з лабораторної роботи повинен містити (дивись лаб. роботу № 2). До звіту необхідно додати файл ArcMap з виконаною роботою.

ЛАБОРАТОРНА РОБОТА № 5

ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ

Мета: виробити уміння і навички під час вирішення прикладних задач з масивами даних за допомогою циклічних обчислювальних процесів у програмі ArcMap за допомогою об'єктно-орієнтованої мови VBA.

Призначення: виконавши роботу, ви навчитеся створювати циклічні обчислювальні процеси для вирішення прикладних завдань.

Підготовка до роботи

1. Що таке цикл, для чого він потрібен?
2. В чому основна відмінність між циклами з передумовою та післяумовою?
3. В яких випадках доцільно використовувати цикли з передумовою, цикли з післяумовою і цикли по лічильнику?
4. Що таке складний цикл та які основні правила його конструювання?

Ключові слова: цикл з лічильником, цикл з передумовою, цикл з післяумовою, математична модель, алгоритм.

Теоретичні відомості

Способи організації циклів. Структура циклічних алгоритмів

Основою розв'язання на комп'ютері прикладних задач є принцип повторення: обчислення виконується багато разів за одними формулами, але при різних початкових даних. Такі процеси називаються циклічними, а частини які повторюються – циклами.

Основу циклічних алгоритмів складають базові алгоритмічні конструкції **цикл**, а основу циклічних програм – оператори циклу.

В циклічних алгоритмах і програмах можна організувати цикл трьома способами:

- цикл з лічильником циклів (з параметром) (рис. 5.1а);
- цикл з передумовою (рис. 5.1б);
- цикл з післяумовою (рис. 5.1в).

Кожному способу відповідає своя структура алгоритму. Типові структури циклічних алгоритмів наведені на рисунку 5.1.

В циклічних алгоритмах можна виділити дві обов'язкової частини: підготовку циклу і цикл. Підготовка циклу (символ 2) включає привласнення початкових значень перемінним, які використовуються для організації циклу та обчислень в циклі. Перемінна, яка використовується для управління циклом, називається параметром циклу.

Цикл включає такі дії: перевірку умови виконання циклу (символ 3) і тіло циклу (символ 4). Тіло циклу – це послідовність дій, які повторюються.

Алгоритми відрізняються тільки способом організації циклу. В алгоритмі з лічильником циклів (рис. 5.1а) як параметр циклу використовується перемінна цілого типу. Лічильник циклів позначається символом «Модифікація» (символ 3). Наприклад, відомо, що цикл виконується n разів. В якості

лічильника циклів використовується перемінна цілого типу, наприклад, i . Під час виконання циклів значення i змінюється від 1 до n з кроком 1 . При $i > n$ цикл завершується. Значення n визначається під час підготовки циклу.

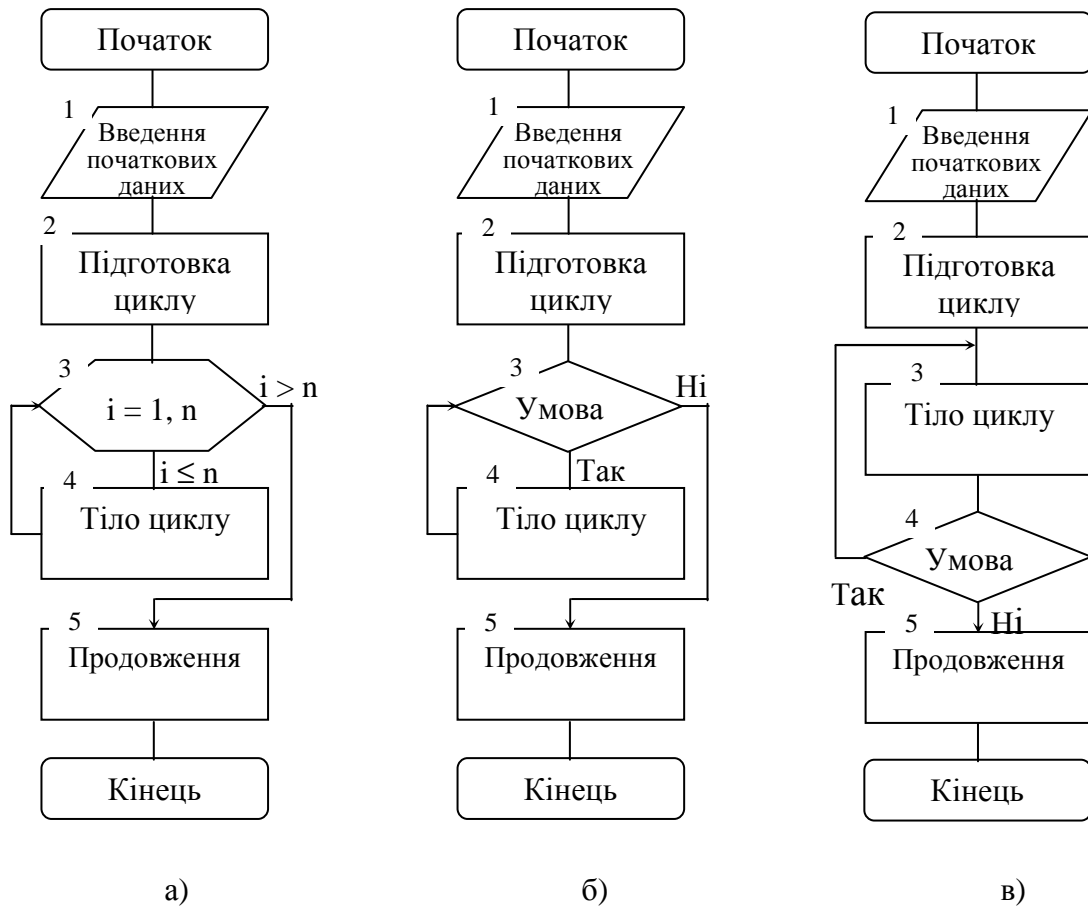


Рисунок 5.1 – Типові структури циклічних алгоритмів

Лічильник циклів застосовується для організації циклів, якщо кількість циклів задано або можна обчислити під час підготовки циклу.

В циклах з предумовою перевірка умови повторення циклу здійснюється перед початком циклу (рис. 5.1б). В умові порівнюються поточні значення параметру циклу із заданим значенням, наприклад, $x < b$. Як параметр циклу може бути змінна будь-якого типу. Цикл виконується доти, доки справедлива умова (**Так**), інакше (**Ні**) цикл завершується. При підготовці циклу необхідно параметру циклу привласнити початкове значення, наприклад, $x = a$. В тілі циклу поточні значення параметру циклу потрібно змінювати, наприклад, $x = x + h$, де h - крок зміни параметру циклу. Інакше умова буде справедлива завжди і цикл буде повторюватися нескінченно.

В циклах з післяумовою перевірка умови повторення циклу здійснюється після циклу (рис. 5.1в). В алгоритмах з лічильником циклів i в циклах

з передумовою цикл може не виконуватися жодного разу, а в циклах з післяумовою цикл виконується хоча б один раз.

Хід роботи

Вправа 1

Постановка задачі

Геодезична компанія отримала великий заказ на проведення геодезичних вишукувань. Оплата праці кожної геодезичної бригади відома. Визначити чи вистачить наявних коштів на оплату праці усіх геодезичних бригад.

Математична модель

Введемо наступні позначення:

- i — номер геодезичної бригади;
- k — кількість геодезичних бригад, які виділені на проведення геодезичних вишукувань;
- R_i — витрати на оплату праці на i -ту геодезичну бригаду;
- S — фонд заробітної плати.

На початку $S=0$. Розподіл заробітної плати на геодезичні бригади проводиться наступним чином. Перевіряють чи вистачає грошей на оплату праці першої бригади ($i = 1$), тобто чи виконується рівняння:

$$S + R_1 \leq Z$$

Якщо вистачає, то ці кошти виплачують ($k = 1$). Потім ті ж дії повторюють з другою геодезичною бригадою ($i = 2$) і так далі.

Загальний фонд заробітної плати на k перших бригад визначається за формулою:

$$S = R_1 + R_2 + \dots + R_k$$

Розрахунок припиняється, якщо розподілена заробітна плата на усі бригади ($k = N$) або фонд заробітної плати скінчився ($k < i$). Останнє рівняння показує, що кількість бригад, на які розподілені гроші, менше ніж кількість бригад, які виконували геодезичні вишукування (на чергову бригаду грошей не вистачило).

Величина k може змінюватися в межах від 0 до N . Випадок $k = 0$ відповідає тому, що грошей не вистачило навіть на першу геодезичну бригаду.

Алгоритм рішення задачі

Рішення задачі складається з наступних етапів (рис. 5.2):

- Введення кількості геодезичних бригад і розмір фонду заробітної плати.
- Установка початкового стану перед розподілом коштів по бригадах.
- Розподіл заробітної плати по бригадам.
- Виведення результату.

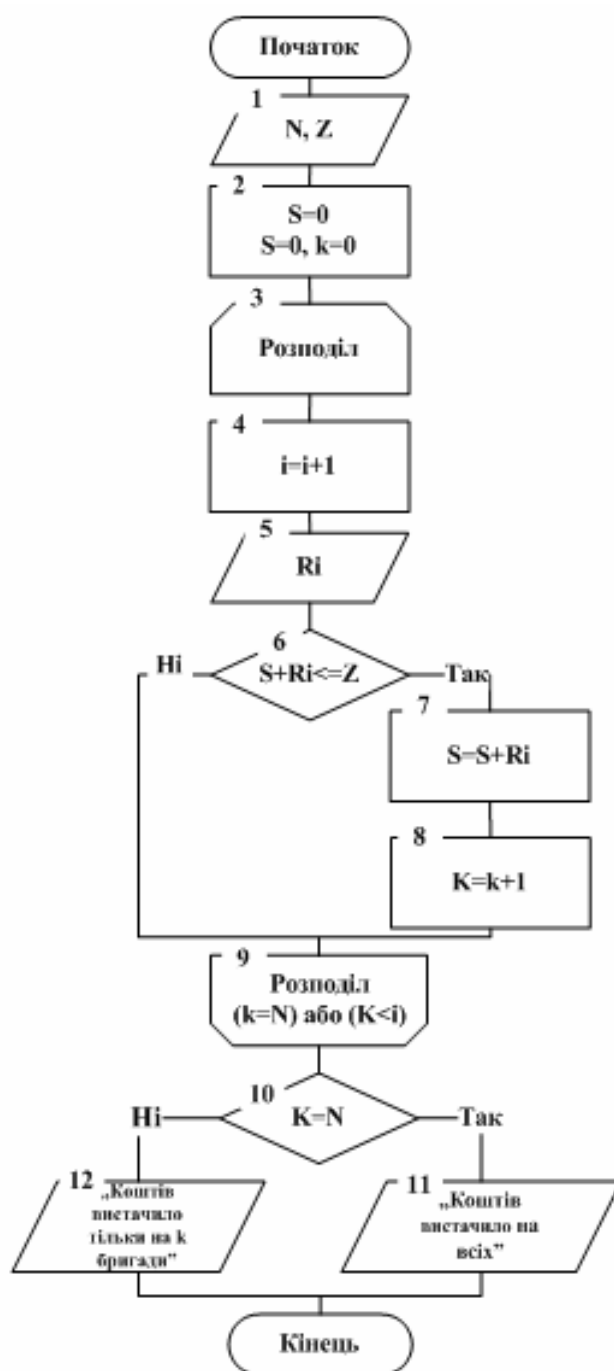


Рисунок 5.2 – Схема алгоритму рішення задачі

Структура даних

В задачі використовуються наступні дані:

- N – кількість бригад, ціле число;
- Z – фонд заробітної плати, дробове число;
- S – загальна сума виплачених коштів, дробове число;
- i – номер бригади, ціле число;
- k – кількість бригад, які виконують геодезичні вишукування;
- R_i – витрата на заробітну плату i -ї бригади, дробове число.

Початкові дані:

N, Z, R_i

Проміжні дані:

i, k, S

Результат:

текстовое повідомлення.

Интерфейс з користувачем

Задача вирішується за допомогою наступної форми (рис. 5.3).

The screenshot shows a Windows application window titled "Заробітна плата" (Wage) with a subtitle "FrmВиплати". The window has a light gray background and a blue title bar. It contains four input fields and three buttons. The first input field is labeled "Кількість бригад" (Number of teams) and has a button "txtN" to its right. The second input field is labeled "Фонд заробітної плати" (Wage fund) and has a button "txtZ" to its right. Below these is a button "Обчислити" (Calculate) with a button "cmdREZ" to its right. The third input field is labeled "Виплати" (Payments) and has a button "txtREZ" to its right.

Рисунок 5.3 – Форма користувача

Введення виплати заробітної плати на кожен бригаду проводиться за допомогою вікна введення **InputBox** (рис. 5.4):

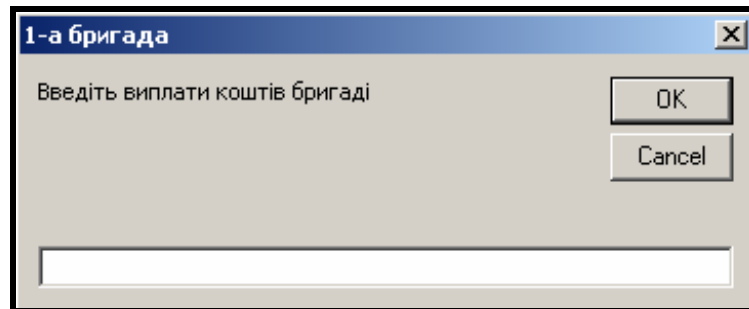


Рисунок 5.4 – Вікно введення заробітної плати

Код програми

Обчислення відбувається після натискання кнопки **Обчислити**. Процедура, яка обробляє цю подію, має наступний вигляд.

```
Option Explicit
Private Sub cmdREZ_Click()

    Dim i As Integer 'номер бригади
    Dim k As Integer 'кількість бригад, які виконують геодезичні вишукування
    Dim Ri As Single 'виплати коштів і-й бригаді
    Dim S As Single 'витрата на заробітну плату і-ї бригади

    'початковий стан
    S = 0
    i = 0
    k = 0

    'розподіл грошей по бригадах
    Do
        i = i + 1
        Ri = CSng(InputBox("Введіть виплати коштів бригаді", i & "-а бригада"))

        If S + Ri <= CSng(txtZ) Then
            S = S + Ri
            k = k + 1
        End If

    Loop Until (k = CInt(txtN)) Or (k < i)

    ' Виведення результату
    If k = CInt(txtN) Then
        txtREZ = "Коштів вистачило на всіх"
    Else
        txtREZ = "Коштів вистачило тільки на " & k & " бригади"
    End If
End Sub
```

Реалізація проекту

- Запустіть ArcMap і створіть новий файл. Натисніть ALT+ F11, щоб перейти в середовище VBA.
- Створіть форму згідно ескізу, який наведено в розділі **Інтерфейс з користувачем**.
- Перейдіть у вікно коду, двічі клацнувши кнопку **Обчислити**, і введіть текст процедури, який представлений у розділі **Код програми**.
- Збережіть документ у своїй папці на жорсткому диску.

Аналіз проекту

Алгоритм рішення задачі реалізований у проекті наступним чином.

Введення вхідних даних проводиться за допомогою текстових полів **txtN** та **txtZ**.

Після клацання по кнопці **Обчислити** починаються обчислення. Вони описані у коді процедури. Спочатку описані перемінні, які потім використовуються в обчисленнях. В іменах цих перемінних не вказані префікси, так як вони полягають з 1-2 букв і за змістом зрозумілий їхній тип.

Перед початком введення витрат заробітної плати на кожен геодезичну бригаду обнуляються наступні величини:

- i – номер поточної геодезичної бригади;
- k – кількість бригад, на які нараховано заробітну плату;
- S – сума фонду заробітної плати.

Для організації розподілу фонду заробітної плати використаний цикл **Do ... Loop**, тому що кількість повторень наперед не відома. Заробітна плата залежить від розміру фонду заробітної плати, кількості всіх геодезичних бригад та витрат на оплату праці кожної бригади.

Оскільки в розподілі фонду заробітної плати бере участь не менше однієї бригади, то використаний цикл з післяумовою (з нижнім закінченням). Умовою виходу з циклу є або повний розподіл фонду заробітної плати на усі бригади $k = \text{CInt}(\text{txtN})$, або, навпаки, на чергову бригаду грошей не вистачило $k < i$. Ці дві умови зв'язано логічною операцією **Or**, тому що достатньо виконання хоча б однієї з цих умов, щоб припинити розподіл заробітної плати та вийти з циклу.

Розподіл фонду заробітної плати на оплату праці чергової геодезичної бригади проводиться наступним чином. Спочатку встановлюється номер цієї бригади $i = i + 1$. Після цього вводиться сума заробітної плати цієї

бригади. Введення здійснюється за допомогою діалогового вікна введення (функція **InputBox**). Це значення перетворюється в числовий тип функцією **CSng**, а потім отриманому значенню присвоюється перемінна **Ri**.

Ri = CSng(InputBox("Введіть виплати коштів бригаді", i & "-а бригада"))

Потім перевіряється, чи вистачить фонду заробітної плати, щоб виплатити гроші наступній бригаді ($S + Ri \leq CSng(txtZ)$). Перевірка проводиться в умовному операторі

If S + Ri <= CSng(txtZ) Then

S = S + Ri

k = k + 1

End If

В ньому вказано, якщо умова виконується, то до заробітної плати бригади додаються витрати на заробітну плату i -ї бригади, а кількість геодезичних бригад, яким виплатили кошти, збільшується на одиницю, тобто **k** стає рівним **i**.

Інакше ці дії не виконуються тому, що в кінці циклу $k < i$. При перевірці умови, записаної після службового слова **Until**, відбудеться вихід з циклу тому що виконалася друга умова.

Loop Until (k = CInt(txtN)) Or (k < i)

Після виходу з циклу перевіряється, з якої саме причини було припинено розподіл коштів, щоб залежно від причини вивести відповідний результат. В операторі **If** перевіряється перша умова.

If k = CInt(txtN) Then

Якщо вона виконується, то виводиться повідомлення про те, що коштів вистачило на всіх.

txtREZ = "Коштів вистачило на всіх"

Останній оператор присвоєння необхідний у тому випадку, коли після отримання першого негативного результату ми бажаємо ще раз клацнути кнопку **Обчислити** і ввести нові дані.

Якщо вказана у операторі **If** умова не виконується (тобто фонду заробітної плати на всі бригади не вистачило), виводиться відповідне повідомлення.

txtREZ = "Коштів вистачило тільки на " & k & " бригади"

При виведенні повідомлення перемінна **k** винесена за лапки, оскільки інакше виводився б символ «**k**», а не значення перемінної.

Виконання програми

Запустіть програму на виконання, клацнувши на вільному місці форми, а потім на кнопці **Run**.

У полі **Кількість бригад** введіть число **3**, а у полі **Фонд заробітної плати** число **10 000** і клацніть кнопку **Обчислити**. У діалоговому вікні, що з'явилося, по черзі введіть числа **2000**, **4000**, **3000**. У полі **Виплати** з'явиться текст **«Коштів вистачило на всіх»**.

Не змінюючи значення в полях **Кількість бригад** та **Фонд заробітної плати**, клацніть кнопку **Обчислити**. У діалоговому вікні, що з'явилося, по черзі введіть числа **7000**, **4000**, **3000**. В полі **Виплати** з'явиться текст **«Коштів вистачило тільки на 1 бригади»**. Невідповідність відмінка в останньому слові свідчить про спрощене ставлення до подання результату. За правилами мови повинно бути написано **«Коштів вистачило тільки на 1 бригаду»**. Спробуйте доопрацювати програму, щоб ліквідувати невідповідність між числівником і відмінком іменника. Якщо вам вдасться виправити цей недолік, то тим самим ви вирішите задачу 4 з розділу «Задачі для самостійного розв'язання».

Завдання для самостійного виконання

1. Додайте на форму малюнки, які будуть відповідати випадкам, коли коштів вистачає на виплату заробітної плати бригадам і коли не вистачає.
2. * Виділіть окремо випадок, коли грошей з фонду заробітної плати точно відповідає потребам на всі виплати. (При цьому повинен відображатися третій малюнок).
3. Змініть властивість текстового поля **Виплати**, щоб його розмір змінювався в залежності від розміру тексту в ньому.
4. * Змініть код програми, щоб ліквідувати невідповідність між числівником і відмінком іменника у полі **Виплати**.
5. Додайте на форму кнопку, за допомогою якої можна завершити роботу програми.

Вправа 2

Постановка задачі

<p>У штатному розкладі підприємства є наступна інформація про N співробітників: ПІБ, посада, оклад. Знайти кількість співробітників, які обіймають задану посаду.</p>
--

Алгоритм рішення задачі

Рішення задачі складається з наступних етапів (рис. 5.5):
Введення кількості співробітників та посади.

1. Установка початкового стану перед підрахунком кількості.
2. Виведення результату.

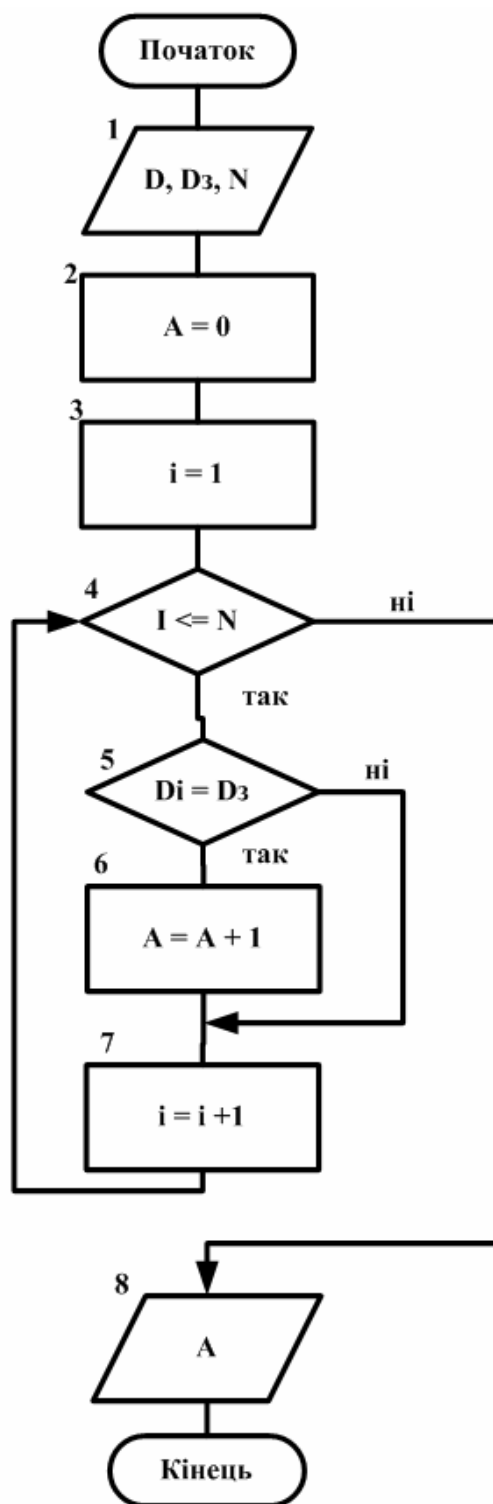


Рисунок 5.5 – Схема алгоритму рішення задачі

Структура даних

В задачі використовуються наступні дані:

- D – масив посад – текстовий рядок;
- F – масив прізвищ – текстовий рядок;
- O – масив окладів – текстовий рядок;
- Dz – задана посада – текстовий рядок;
- N – кількість співробітників – ціле число;
- A – лічильник – ціле число;
- i – номер співробітника – ціле число.

Интерфейс з користувачем

Задача вирішується за допомогою наступної форми (рис. 5.6).



Рисунок 5.6 – Форма користувача

Введення прізвища, посади та заробітної плати на кожного співробітника виконується за допомогою вікон введення InputBox (рис. 5.7).

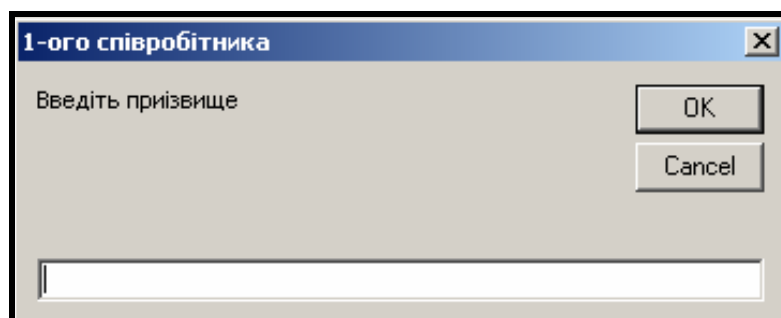


Рисунок 5.7 – Вікно введення прізвища співробітника

Код програми

Обчислення відбувається після клацання на кнопці **Обчислити**. Процедура, яка обробляє цю подію, має наступний вигляд.

Option Explicit

Private Sub cmdREZ_Click()

'Опис пермінних

Dim A As Integer

Dim i As Integer

Dim Oi As Single

Dim Fi As String

Dim Di As String

Dim Dz As String

A = 0

'Введення прізвищ, посад, заробітних плат

For i = 1 To CInt(txtN)

Fi = CStr(InputBox("Введіть прізвище", i & "-ого співробітника"))

Di = CStr(InputBox("Введіть посаду", i & "-ого співробітника"))

Oi = CSng(InputBox("Введіть заробітну плату", i & "-ого співробітника"))

'Визначення заданої посади

If Di = CStr(txtDz) Then

A = A + 1

End If

Next i

'Введення кількості співробітників, які обіймають задану посаду

txtREZ = Format(CStr(A) & " співробітників")

End Sub

Реалізація проекту

- Запустіть ArcMap і створіть новий файл. Натисніть ALT+ F11, щоб перейти до середовища VBA.
- Створіть форму згідно ескізу, який наведено в розділі **Інтерфейс з користувачем**.
- Перейдіть до вікна коду, двічі клацнувши кнопку **Обчислити**, введіть текст процедури, який представлений в розділі **Код програми**.
- Збережіть документ у своїй папці на жорсткому диску.

Аналіз проекту

Алгоритм рішення задачі реалізований у проекті наступним чином.

Введення вхідних даних проводиться за допомогою текстових полів **txtN** та **txtDz**.

Після клацання по кнопці **Обчислити** починаються обчислення. Вони описані в коді процедури. Спочатку описані перемінні, які потім використовуються в обчисленнях. Перед початком введення прізвищ, посад та заробітних плат кожного співробітника обнуляється лічильник.

Виконання програми

Запустіть програму на виконання, клацнувши на вільному місці форми, а потім по кнопці **Run**.

В полі **Кількість співробітників** введіть число **3**, а в полі **Посада** вкажіть бажану посаду і клацніть кнопку **Обчислити**. У діалоговому вікні, що з'явилося, по черзі введіть прізвище, посаду та заробітну плату кожного з трьох співробітників. В полі **txtREZ** з'явиться кількість співробітників, які обіймають вказану посаду.

Завдання для самостійного виконання

1. * Визначіть сумарний фонд заробітної плати усіх співробітників.
2. * Визначіть середній оклад усіх співробітників.
3. Додайте на форму кнопку, за допомогою якої можна завершити роботу програми.

Контрольні питання

1. Які два види циклічних структур Ви знаєте? Опишіть їх відмінності.
2. Що відбувається з циклом, якщо значення кроку буде негативне число?
3. Опишіть як «працює» цикл, якщо кінцеве значення лічильника циклу менше, ніж початкове?

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений на ПК, роздрукований на аркушах формату А4. Звіт повинен містити (дивись лаб. роботу № 2). До звіту необхідно додати файл, виконаний у програмному забезпеченні ArcMap з виконаною роботою.

ЛАБОРАТОРНА РОБОТА № 6

ВИКОРИСТАННЯ ПІДПРОГРАМ ТА ФУНКЦІЙ

Мета: виробити уміння і навички розробки процедур та модулів за допомогою об'єктно-орієнтованої мови VBA у програмі ArcMap.

Призначення: виконавши роботу, ви навчитесь створювати обчислювальні процеси з використанням процедур та модулів для вирішення громіздких прикладних завдань.

Підготовка до роботи

1. Дайте визначення процедурі. Яким чином оформлюються процедури у VBA?
2. Призначення функцій. Який вид мають функції?
3. Що представляють собою модулі?
4. Які типи модулів ви знаєте?
5. Дайте визначення класу. Для чого вони використовуються у VBA?

Ключові слова

Процедура, функція, модуль, клас, аргумент, ім'я процедури, об'єктно-орієнтоване програмування.

Теоретичні відомості

Процедура – це сукупність операторів, що виконують певні дії. Процедури мають стандартне оформлення:

Public {[**Private**],[**Static**]} **Sub** *Name* (Список параметрів)
тіло процедури
End Sub

Public – глобальна процедура, доступна для усіх інших процедур у всіх модулях проекту;

Private – процедура модуля, доступна для усіх інших процедур в даному модулі;

Static – службове слово, яке говорить про те, що локальні змінні процедури зберігаються в проміжках часу між викликами цієї процедури

Name – ім'я процедури, що задовольняє стандартним правилам написання імен у VBA. Ім'я процедури обробки події складається з імені об'єкту та імені події.

Список аргументів – список формальних параметрів (аргументів) процедури **Sub** (імен змінних, які повинні бути передані в процедуру під час звернення до неї).

Синтаксис елементу структури **Список аргументів** наступний:

[Optional] [ByVal, ByRef] [ParamArray] Ім'я змінної As Тип

Optional – ключове слово, вказує, що аргумент не є обов'язковим. Усі аргументи, описані як **Optional**, повинні бути типу **Variant**.

ByVal – вказує на те, що цей аргумент передається по значенню.

ByRef – вказує на те, що цей аргумент передається по посиланню на його адресу в пам'яті. Опис **ByRef** використовується за замовчуванням.

ParamArray – використовується тільки як останній елемент у списку аргументів для вказівки на те, що кінцевим аргументом у списку параметрів процедури є масив значень, описаний як тип **Variant**, тобто це слово дозволяє задавати довільну кількість аргументів у процедурі.

Функції багато в чому схожі на процедури. Існує лише одна принципова відмінність – під час виклику вони повертають значення. Функція одержує один або декілька об'єктів даних, званих аргументами, і виконує з ними деякі дії. Їхній результат повертається функцією.

Функції мають вигляд:

Function Ім'я_Функції [(Список аргументів)] **As** тип
тіло функції
End Function

Модулі представляють собою текстові ASCII-файли з програмним кодом. В них зручно групувати взаємопов'язані процедури, які можуть використовуватися в програмі.

Код проекту може складатися з безлічі програмних модулів. **Класи** представляють собою основні будівельні блоки об'єктно-орієнтованого програмування – моделі, в якій програма описується у вигляді сукупності об'єктів. Клас не тільки дозволяє виділити частину функціональних засобів програми в окремий об'єкт, але й розширює можливості базових модулів. Ви можете захистити одні із фрагментів програми, а інші надати в розпорядження програми. Цей процес називається *інкапсуляція*.

Оскільки модулі розміщуються в окремих файлах, їх можна включати відразу в декілька проектів. Таким чином, з'являється можливість повторно використовувати написаний код.

Модуль класу – це спеціальний тип модулю, який використовується під час створення класів, що розробляються програмістом для вирішення своїх задач.

В VBA передбачена можливість створення призначених для користувача об'єктів, які є екземплярами класів. Всі об'єкти одного й того ж класу використовують однакові методи у відповідь на однакові повідомлення. Класи конструюються в модулях класів, які створюються командою **Insert – Class Module** і записуються у вікні, що з'явилося. Натиснувши клавішу **F4**, можна вибрати ім'я класу.

Хід роботи

Вправа 1

Постановка задачі

Для зручності користувачів під час роботи з картою необхідно додати процедуру, за допомогою якої вони зможуть збільшувати потрібний фрагмент карти та роздивлятися його.

- Запустіть **ArcMap** і відкрийте вправу **ex06a.mxd**. Коли карта відкриється ви побачите необхідні шари та панель інструментів **Crime Analysis** (Аналіз злочинів) (рис. 6.1).

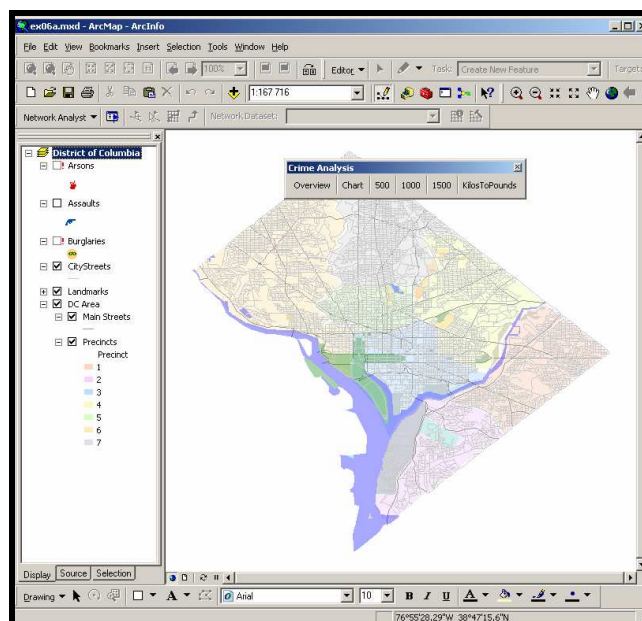


Рисунок 6.1 – Вікно програми ArcMap з відкритою вправою

- Додайте до інтерфейсу програми панелі **Draw**, **Standard** та **Tools**. Викличте контекстне меню на кнопці **Overview**, яка розташована на панелі **Crime Analysis** і виберіть команду **ViewSource**.

У вікні коду **ThisDocument** ви побачите обрамляючі рядки для події клацання ЛКМ: **Overview click ()** (рис. 6.2).

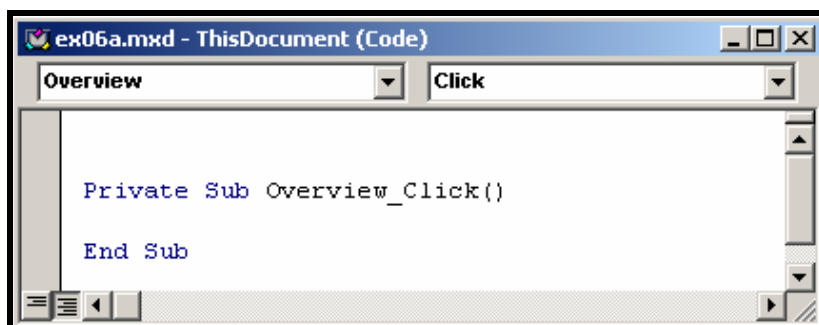


Рисунок 6.2 – Процедура для кнопки **Overview**

- Між рядками початку та закінчення процедури необхідно вписати код, за допомогою якого буде викликана процедура **CreateOverviewWindow**. Для цього процедуру потрібно імпортувати.

- За допомогою контекстного меню у вікні **Project** викличте пункт **Import File**. Вкажіть у списку типів файлів (**Files of type**) «Всі файли» (**All files**). Потім перейдіть до папки **...\ArcObjects\Data\Samples\ExploringArcObjects** і виберіть файл **CreateOverviewWindow.txt** (рис. 6.3).

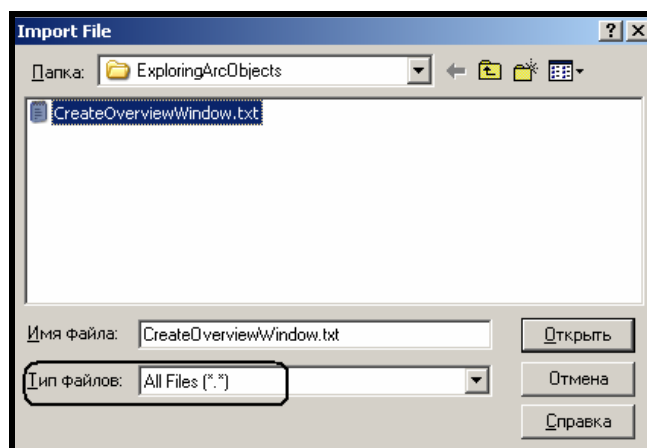


Рисунок 6.3 – Вибір файлу з процедурою

Новий код буде добавлений у ваш проект у вигляді нового модулю – **Module1**, який ви можете відкрити. В даному модулі міститься процеду-

ра, в якій ви при бажанні можете розібратися. Даний код складається з трьох частин:

- Перша частина представляє собою оголошення перемінних за допомогою оператора **Dim**. Зверніть увагу, що перемінні, які тут описуються мають об'єктні типи. Тут ви зустрінетеся з об'єктами бібліотеки **ArcObjects**.

- У другій частині цим перемінним присвоюються щойно створені об'єкти за допомогою оператора **Set**. З цим оператором докладніше ми познайомимося пізніше.

- В третій частині коду змінюються деякі властивості створених об'єктів, щоб задовольнити естетичним і ергономічним вимогам додатка.

У вікні властивостей (**Properties**) змініть ім'я модулю **Module1** на ім'я **CrimeAnalysisTasks** (рис. 6.4).

В інших вправах цієї лабораторної роботи ви додасте ще декілька процедур в цей модуль.

ПРИМІТКА. Взагалі можна кожену процедуру зберігати в її власному модулі. Але логичніше якщо процедури, які вирішують подібні задачі зберігати в одному модулі.

Процедури обробника подій елементів управління **UIControls** (таких як кнопка **Overview**) зберігаються у модулі **ThisDocument**. Це означає, що процедура з одного модулю повинна бути викликана в іншому модулі. Отже, ключове слово, впливаюче на область видимості (розміщується перед ключовим словом початку процедури – **Sub**) повинне бути **Public**, що означає відкритий доступ. Обмежений доступ означає слово **Private**. Процедури з обмеженим доступом можуть викликатися тільки всередині свого модулю.

- Активізуйте вікно модуля **ThisDocument**. Додайте в процедуру обробника події – клацання ЛКМ на кнопці **Overview** наступний рядок (рис. 6.5).

Call CreateOverviewWindow

- Закрийте редактор VBA.

Перевірка роботи імпортованої процедури

Існує декілька точок недавніх крадіжок зі зломом недалеко від коледжу **Holy Name**. Ви можете тестувати вашу процедуру для перевірки області навколо коледжу.

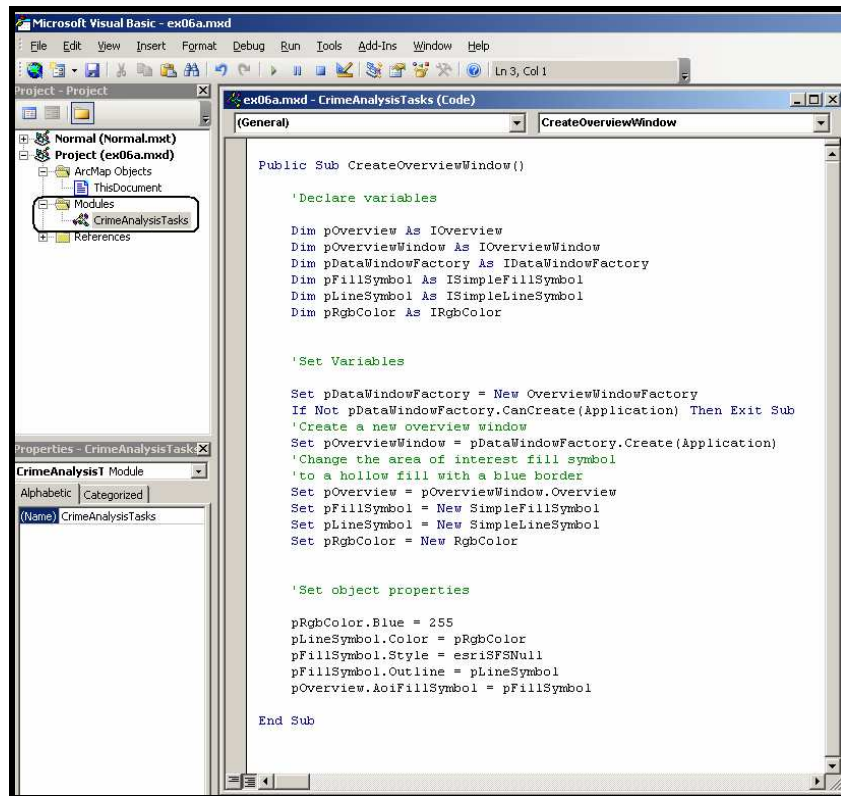


Рисунок 6.4 – Вигляд процедури **CrimeAnalysisTasks**

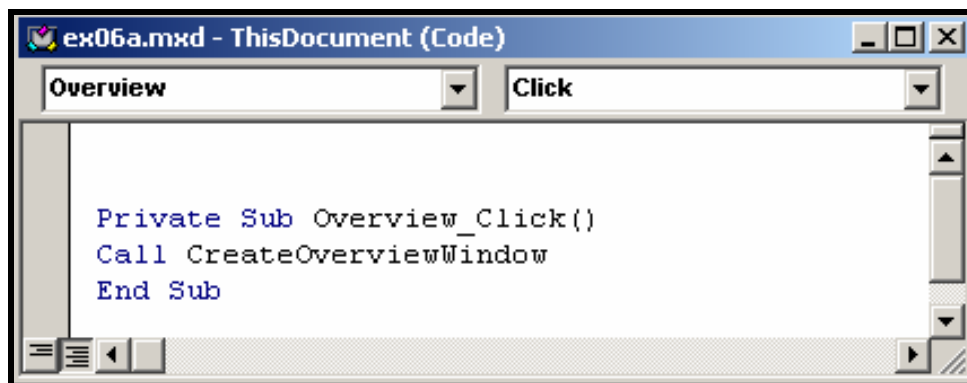


Рисунок 6.5 – Процедура для кнопки **Overview**

Включіть шари **LandMarks** та **Burglaries**. Виберіть з меню **Bookmarks** команду **Holy Name College**. Оскільки ми збільшили масштаб, важко сказати в якому місці округу знаходиться вибраний об'єкт.

За допомогою кнопки **Overview** панелі інструментів **Crime Analysis** можна переміщати або змінювати розміри вікна **Overview**. В цьому вікні ви побачите синій прямокутник, який відображує те, що збільшена в масштабі ділянка знаходиться на північному сході округу. Щоб переглянути злочини в інших частинах міста, необхідно переміщати або змінювати розміри синього прямокутника у вікні **Overview**. Всі ці зміни будуть відо-

бражатися в головному вікні **ArcMap** (рис. 6.6). Поекспериментуйте. Закрийте вікно огляду і збережіть свою роботу.

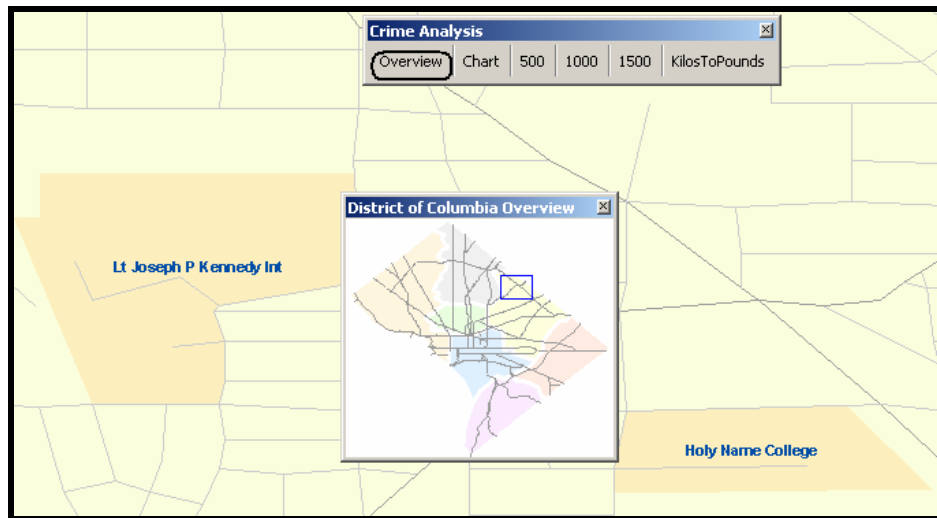


Рисунок 6.6 – Перегляд карти за допомогою вікна **Overview**

Вправа 2

Постановка задачі

Для зручності користувачів під час аналізу шарів карти з позначками підпалів по округах необхідно додати процедуру, за допомогою якої вони зможуть візуалізувати ці дані у діаграму.

- Запустіть **ArcMap** і відкрийте вправу **ex06b.mxd**. Коли карта відкриється ви побачите необхідні шари та панель інструментів **Crime Analysis** (Аналіз злочинів).
- За допомогою контекстного меню на кнопці **Chart** викличте команду **ViewSource**.
- У вікні **ThisDocument** ви побачите обрамляючі рядки для події клацання: **Chart click ()**. Як і в попередній вправі необхідно імпортувати процедуру за адресою **...\ArcObjects\Data\Samples\ArcObjectsOnline\CreateNewChart.txt**. Новий код буде доданий у ваш проект у вигляді нового модулю **Module1**.
- Відкрийте щойно імпортований модуль і скопіюйте процедуру з нього в модуль, створений в попередній вправі **CrimeAnalysisTasks**.
- Видаліть **Module1**, використовуючи контекстне меню і команду **Remove**. На запитання про експортування модулю відповідайте **NO**.

- В модулі **CrimeAnalysisTasks** переробіть додану процедуру так, щоб у неї був аргумент. Додайте аргумент **strTitle** всередині круглих дужок після імені процедури, так щоб перший рядок процедури мав такий вигляд (рис. 6.7):

Public Sub CreateNewChart (strTitle As String)

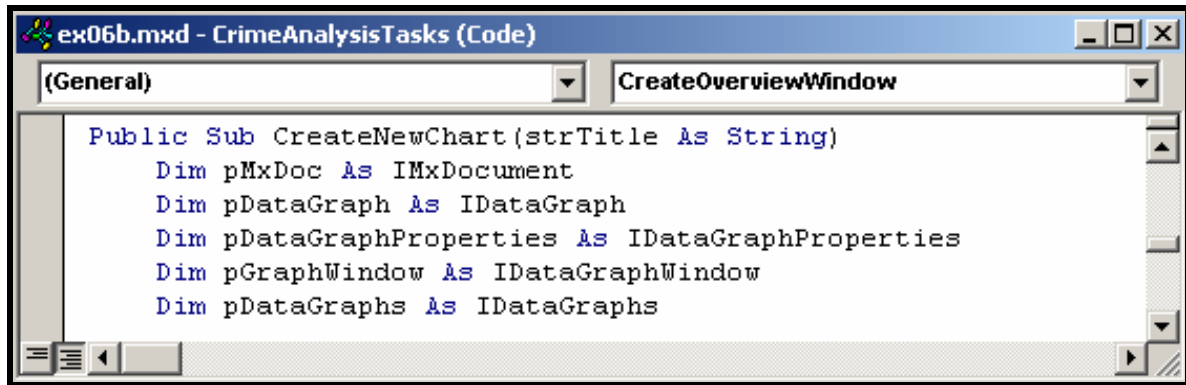


Рисунок 6.7 – Вигляд процедури CrimeAnalysisTasks

Знайдіть та відредагуйте наступні рядки коду, які встановлюють властивості:

- Title** діаграми.

Початковий рядок має бути таким:

pDataGraphProperties.Title = «My Chart»

- Змініть «**My Chart**» на перемінну **strTitle**. Рядок коду буде виглядати наступним чином:

pDataGraphProperties.Title = strTitle

Перевірте, що ім'я перемінної ви вставили без лапок.

- Активізуйте вікно **ThisDocument** і зайдіть у процедуру **Chart_click ()**. Опишіть змінну для заголовка діаграми. Для того, щоб запитувати заголовок діаграми у користувача необхідно скористатися процедурою **InputBox**.

- Для виклику процедури побудови діаграми скористайтесь процедурою **CreateNewChart** (рис. 6.8).

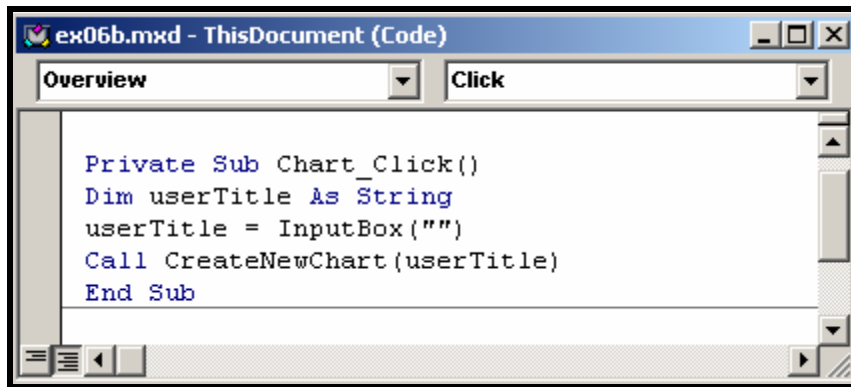


Рисунок 6.8 – Вигляд процедури **Chart_click ()**

- Закрийте вікно редактора VBA.
- Виділіть шар **Precinsts**. Натисніть на кнопку **Chart** на панелі інструментів **Crime Analysis**. В діалоговому вікні напишіть назву майбутньої діаграми «**Підпали по округах**» і клацніть на кнопці **OK** (рис. 6.9).

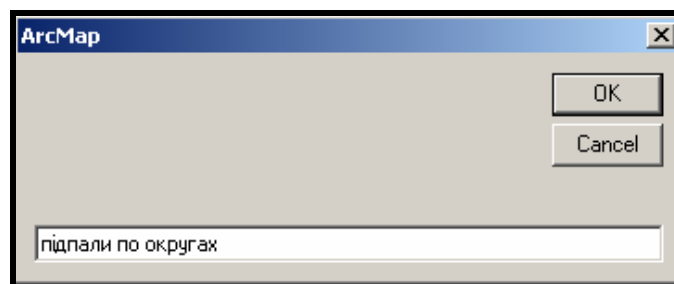


Рисунок 6.9 – Вигляд процедури CrimeAnalysisTasks

Повинна з'явитися діаграма, яка представлена нижче (рис. 6.10). З цієї діаграми видно, що в окрузі 4 було 20 підпалів, а в окрузі 6 не зафіксовано жодного.



Рисунок 6.10 – Вигляд створеної діаграми

- Закрийте діаграму і збережіть свою роботу.

Вправа 3

Постановка задачі

Для зручності корисувачів під час аналізу карти необхідно додати процедуру, за допомогою якої вони зможуть візуалізувати за допомогою показу на кірті буферних зон навколо місць загоряння під час дослідження цих випадків.

- Запустіть **ArcMap** і відкрийте вправу **ex06c.mxd**. Коли карта відкриється ви побачите необхідні шари та панель інструментів **Crime Analysis** (Аналіз злочинів). Включіть шар **Arsons**.
- За допомогою контекстного меню на кнопці **500** викличте команду **ViewSource**.
- У вікні **ThisDocument** ви побачите обрамляючі рядки для події клацання: **Buffer500 click ()**. Як і в попередній вправі необхідно імпортувати процедуру за адресою **...\ArcObjects\Data\Samples\ArcObjectsDeveloperHelp\BufferFeatures.txt**.
- Новий код буде доданий до вашого проекту у вигляді нового модулю **Module1**.
- Відкрийте щойно імпортований модуль і скопіюйте процедуру **BufferFeatures** з нього в модуль, створений в попередній вправі **CrimeAnalysisTasks**.
- Видаліть **Module1**, використовуючи контекстне меню і команду **Remove**. На запитання про експортування модулю відповідайте **NO**.
- У модулі **CrimeAnalysisTasks** переробіть додану процедуру так, щоб у неї був аргумент.
- Додайте аргумент **strBufferDistance** всередину круглих дужок після імені процедури, так щоб перший рядок процедури мав такий вигляд:
Public Sub BufferFeatures (strBufferDistance As String)
- Знайдіть рядок коду, в якому оголошується дана перемінна і видаліть її:

Dim strBufferDistance As String

- Нижче у програмному коді знайдіть наступний рядок:

strBufferDistance = InputBox("Enter Distance:", "Buffer")

Цей рядок запрошує значення розміру буфера у користувача. У вас ця процедура буде викликатися з тими значеннями, які вже задані. Тому даний рядок не потрібен. Видаліть цей рядок коду.

- Активізуйте вікно **ThisDocument**.

Далі потрібно створити обробники для подій клацання ЛКМ по трьох кнопках: 500, 1000 і 1500, щоб викликати дану процедуру з наперед зарезервованими значеннями величинами буфера.

- Між обрамляючими рядками події **Buffer500_click()** вставте рядок виклику процедури, що створює буфер із значенням величини 500.

Call BufferFeatures ("500")

Тут 500 передається як текстовий рядок і тому береться в лапки.

- З лівого випадаючого списку вікна **ThisDocument** виберіть об'єкт **Buffer1000**. В обрамляючих рядках для події **Buffer1000_click()** вставте рядок виклику тієї ж процедури але зі значенням 1000:

Call BufferFeatures ("1000")

- Аналогічно зробіть обробник події для кнопки 1500.
- Закрийте вікно редактора VBA.

Протестуйте кнопки 1500, 1000, 500 на панелі Crime Analysis

Використаємо кнопки для створення буферів та дослідження випадків підпалів поблизу університету **Howard**.

- Виберіть з меню **Bookmarks** команду **Holy Name College**. Переведіть карту у режим редагування. Потім виділіть університет **Howard** (рис. 6.11).
- На панелі інструментів **Crime Analysis** клацніть послідовно на кнопках 1500, 1000, 500.
- Ви побачите побудовані буфери. За допомогою команди **Select All Elements** меню **Edit** виділіть усі буферні зони (рис. 6.12).

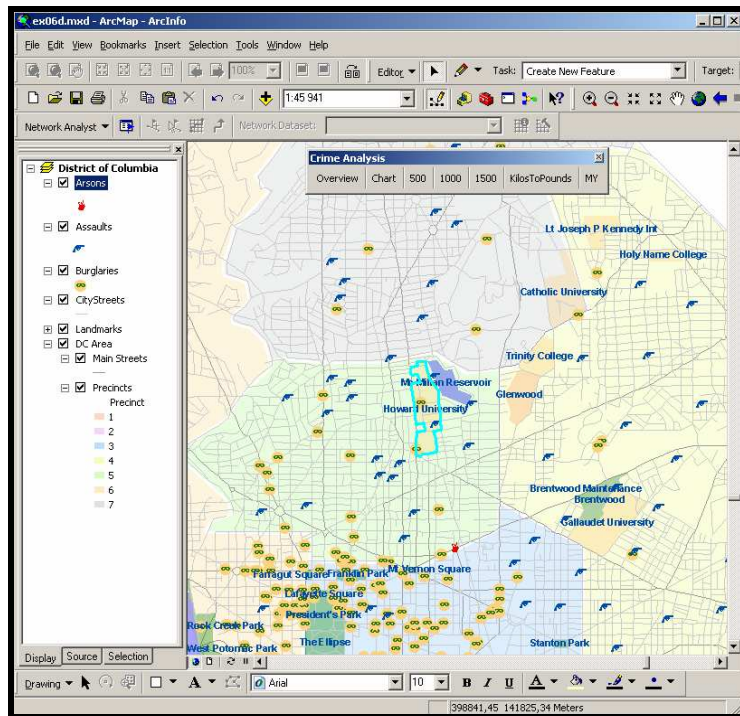


Рисунок 6.11 – Карта з виділеною територією університету Howard

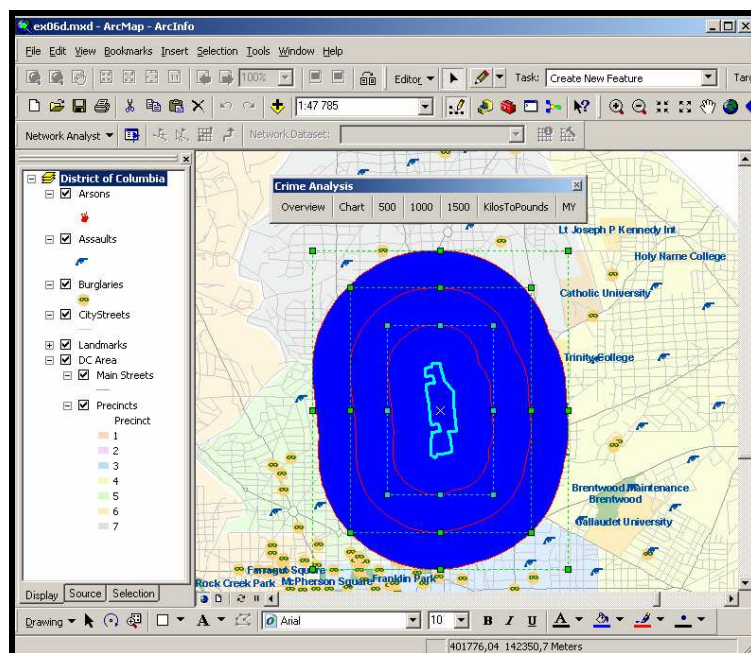


Рисунок 6.12 – Виділені буферні зони

Викличте контекстне меню на будь-якій буферній зоні та виберіть команду **Properties**. В діалоговому вікні **Selected elements** на вкладці **Symbol** клацніть по кнопці **Change Symbol** (рис. 6.13).

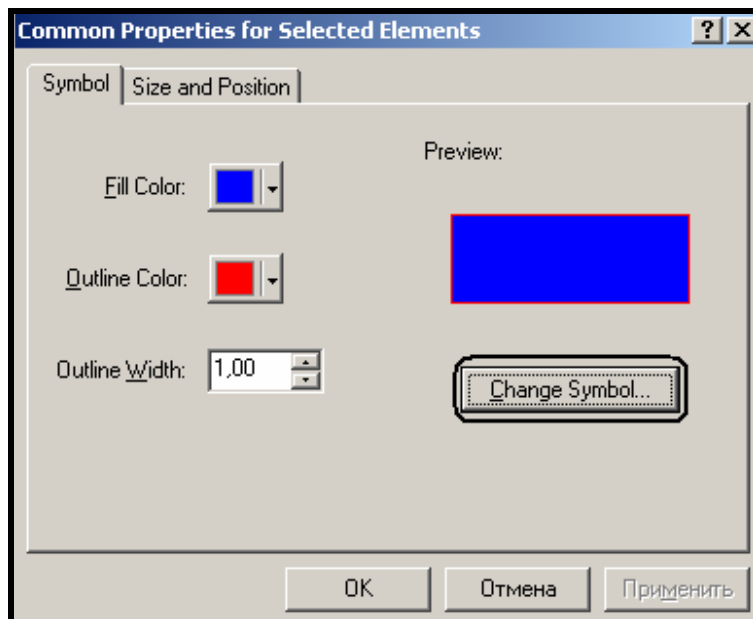


Рисунок 6.13 – Вікно властивостей вибраних буферних зон

- У вікні вибору символів за допомогою полоси прокрутки перейдіть вниз та двічі клацніть на символі, який називається **Crime Reporting Sector**. Натисніть кнопку **OK** (рис. 6.14).

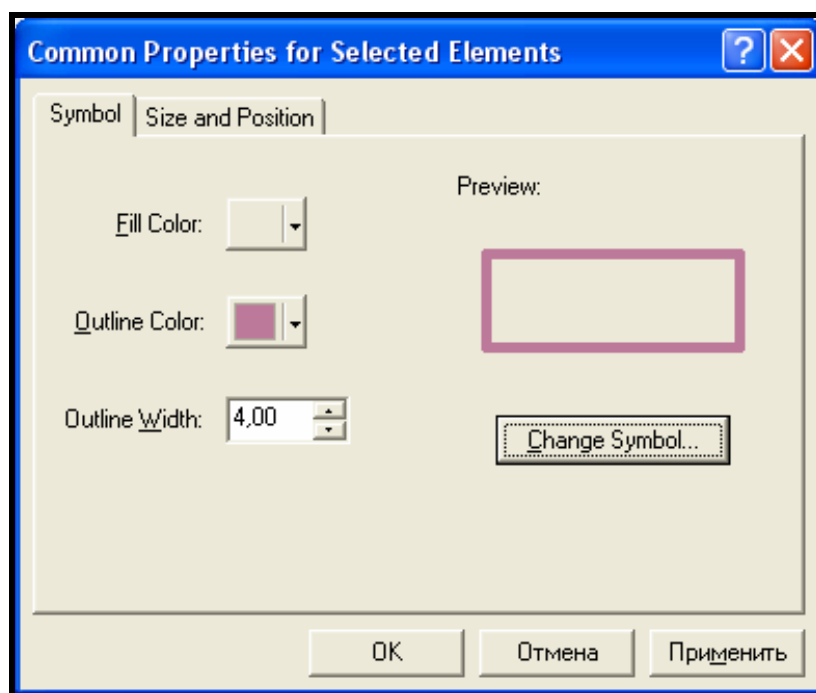


Рисунок 6.14 – Вигляд процедури CrimeAnalysisTasks

Декілька підпалів дійсно трапилося всередині виділених буферних зон (рис. 6.15). Збережіть свою роботу.

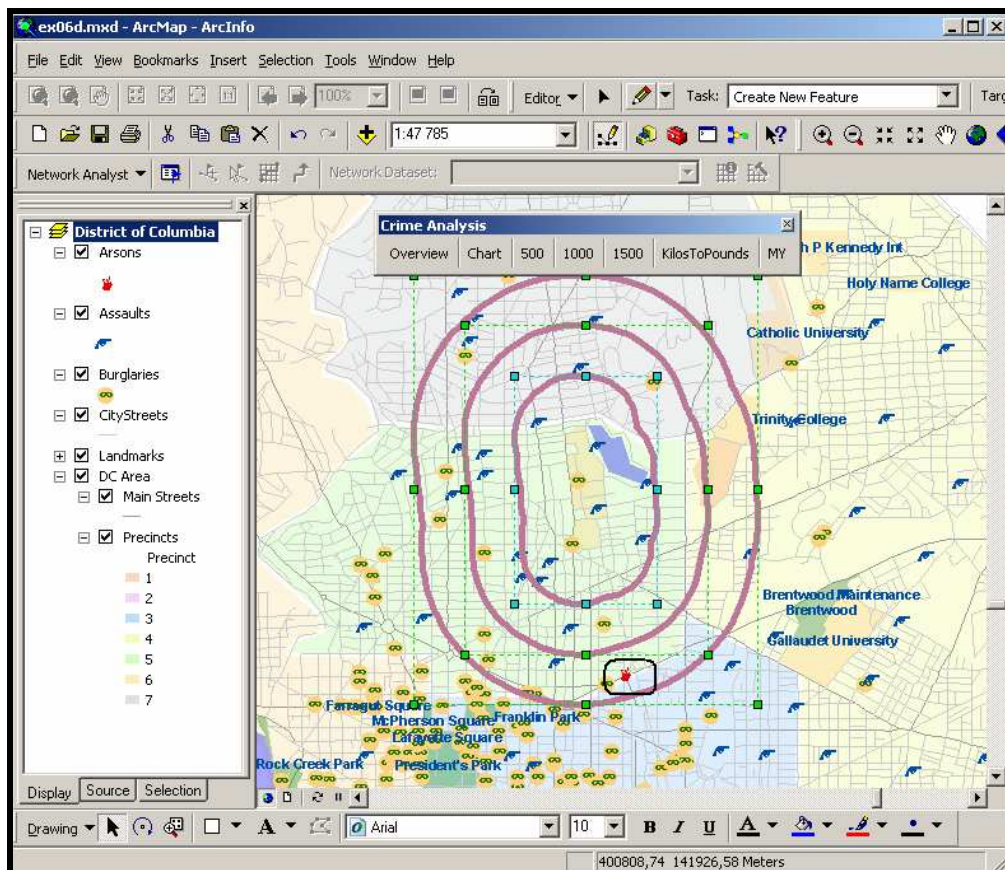


Рисунок 6.15 – Зображення підпалів на карті всередині буферних зон

Вправа 4

Постановка задачі

Для зручності користувачів під час переведення з одних одиниць виміру маси в інші необхідно додати процедуру, за допомогою якої вони зможуть конвертувати дані.

- Запустіть **ArcMap** і відкрийте вправу **ex06d.mxd**. Коли карта відкриється ви побачите необхідні шари та панель інструментів **Crime Analysis** (Аналіз злочинів). Включіть шар **District**.
- Відкрийте редактора VBA (меню **Tools / Macros / Visual Basic Editor**).
- Відкрийте модуль, який містить добавлені процедури **CrimeAnalysisTasks**.
- При активному вікні модулю додайте нову процедуру за допомогою меню **Insert / Procedure**, вказавши наступні параметри:
Ім'я (Name) – **KilogramToPound**.
Тип (Type) – **Function**.

Область видимості (Scope) – **Public** (рис. 6.16).

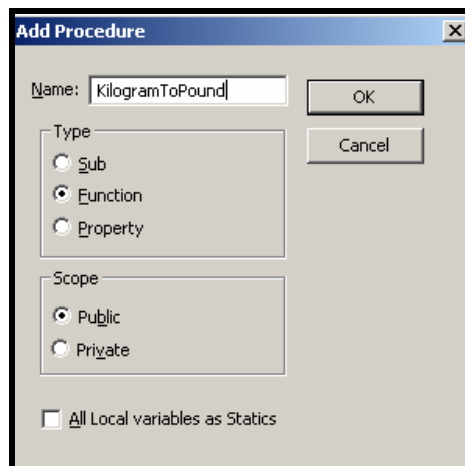


Рисунок 6.16 – Вікно додавання процедури

- У списку аргументів доданої функції наберіть **dblKilos As Double**. Тим самим ви визначили тип значення, що повертається, як подвійна точність.

Тобто перший рядок вашої функції повинен мати вигляд:

Public Function KilogramToPound (dblKilos As Double) As Double

Введіть формулу для перерахунку:

KilogramToPound=dblKilos *2.2046

Таким чином, ваша функція цілком закінчена і повинна виглядати так (рис. 6.17):

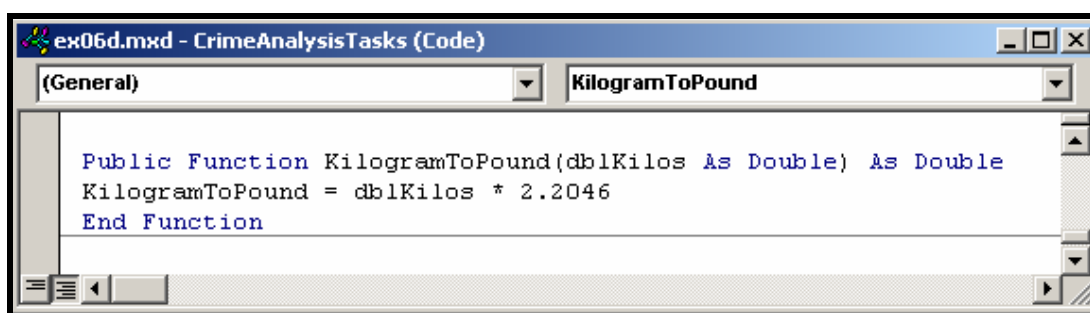


Рисунок 6.17 – Код процедури KilogramToPound

- Поверніться в головне вікно **ArcMap** і за допомогою контекстного меню на кнопці **KilosToPounds** викличте вікно **ViewSource**.
- Використайте вікно вводу даних **InputBox** для введення значення кілограмів для перерахунку, необхідних користувачеві, до програми.

Будь-яке значення, введене користувачем повинно бути переадресовано до функції **KilosToPounds**.

- Всередині обрамляючих рядків процедури обробника події **KilosToPounds_click()** необхідно описати перемінну для введеного користувачем значення кілограмів та присвоїти їй значення за допомогою функції **InputBox**.

Dim userKilos As Double

userKilos = InputBox ("Введіть кількість кілограмів")

Тепер у вас є необхідне значення для виклику функції **KilosToPounds**.

- Опишіть перемінну, яка прийме розраховане значення і присвойте їй значення за допомогою функції **KilogramToPound**:

Dim userPounds As Double

userPounds = KilogramToPound(userKilos)

Тепер необхідно додати рядок, який буде доводити до користувача конвертоване значення фунтів (рис. 6.18):

**MsgBox userKilos & " кг будуть мати " & _
userPounds & " фунтів"**

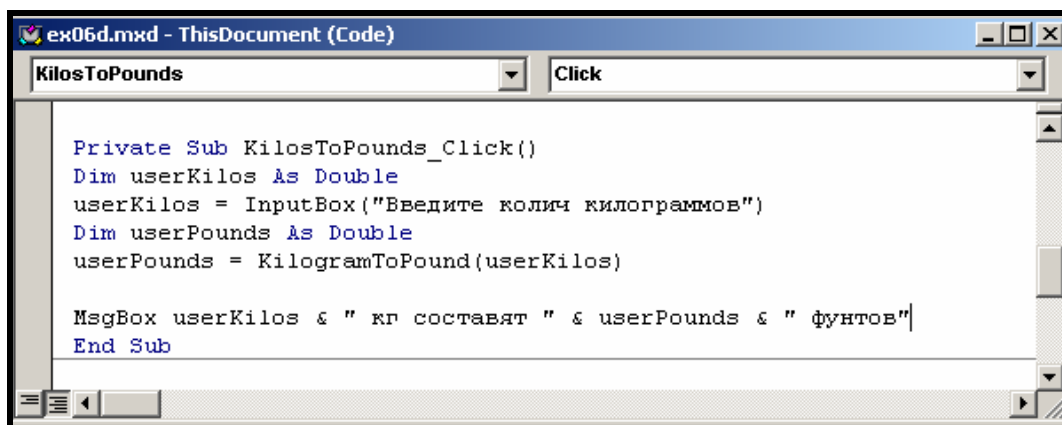


Рисунок 6.18 – Код процедури обробника події **KilosToPounds_click()**

Тепер кнопка **KilosToPounds** готова для тестування.

- Закрийте редактор VBA.

- Клацніть на кнопці **KilosToPounds** та у вікні **InputBox**, що з'явився, введіть 14. Натисніть кнопку **ОК**. Результат повинен скласти 30,3644 фунтів. Клацніть по кнопці **ОК**.
- Збережіть свою роботу.

Завдання для самостійного виконання

1. Внесіть зміни у вікно повідомлення, яке з'являється під час клацання на кнопці **KilosToPounds**, щоб воно мало назву у рядку імені «**Інформаційне повідомлення**» і мало дві кнопки «**ОК**» та «**Відмінити**» (функція **MsgBox**).
2. Додайте на панель інструментів **Crime Analysis** кнопку, за допомогою якої буде здійснено переведення введеного числа з одних одиниць вимірювання в інші (завдання для самостійного виконання лабораторної роботи № 3. Розгалуження не потрібне). Наприклад, з гривні у долари. Код програми повинен бути створений за допомогою модулю (дайте йому своє ім'я) та функції. Вікно повідомлення повинно мати такий вигляд, який вимагається у завданні № 1.

Контрольні питання

1. До яких недоліків призводить створення довгих кодів та яким чином їх можна подолати використавши підпрограми.
2. Перерахуйте типи підпрограм і їхні особливості.
3. Чи може процедура мати аргументи?
4. Чи може функція не мати аргументів?
5. Чи може процедура повертати значення?
6. Чи може функція не повертати значення?
7. Які значення повертає функція **MsgBox**?
8. Які значення повертає функція **InputBox**?
9. Скориставшись довідкою знайдіть функції конвертації типів. Їхнє призначення? Аргументи і значення, що повертаються? Перерахуйте деякі з них.

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений на ПК, роздрукований на аркушах формату А4. Звіт повинен містити (дивись лаб. роботу № 2). До звіту необхідно додати файл, виконаний у програмному забезпеченні ArcMap з виконаною роботою.

ЛАБОРАТОРНА РОБОТА № 7

РОЗРОБКА ВЛАСНИХ ОБ'ЄКТІВ

Мета: виробити уміння і навички розробки власних об'єктів в об'єктно-орієнтованій мові VBA у програмі ArcMap.

Призначення: виконавши роботу, ви навчитеся створювати обчислювальні процеси з використанням власних об'єктів для вирішення громіздких прикладних завдань.

Підготовка до роботи

1. Що таке об'єкти та моделі об'єктів?
2. Що таке об'єкти-набори. Для чого вони використовуються?
3. Яким чином створити зв'язок моделі об'єктів з інтерфейсом користувача?
4. Як створити колекцію своїх об'єктів?

Ключові слова

Об'єктно-орієнтоване програмування, об'єкт користувача, мова UML, інтерфейс.

Теоретичні відомості

Програмісти Microsoft вже створили всі необхідні класи для роботи з об'єктами VBA, такі як форми і елементи управління. А програмісти ESRI, у свою чергу вже написали класи для ГІС-об'єктів, таких як карти і шари. Проте для вирішення специфічних завдань потрібно створювати свої класи.

Уявіть собі, що ви працюєте у муніципальній міській компанії, яка створює додаток для податкового управління з інвентаризації. Він буде корисний для моделі міських кварталів в ArcMap. Але VBA та ArcMap не мають об'єкта **Квартал (Parcel)**. Можна створити свій клас кварталів з класу **Parcel Feature** в базі геоданих і запитувати його властивості. Але не існує програмного об'єкта, який зветься квартал. Тому не можна написати код для запиту значення кварталу (його номер або ранг):

MsgBox myParcel.Value

Не можна також написати код для отримання кодування зони кварталу:

MsgBox myParcel.Zoning

Тому, при частому використанні доцільно створити власний клас.

Перед тим, як програмувати потрібно спланувати, які об'єкти та методи будуть входити до цього класу. Перш за все треба висловити свої думки на папері. Але рано чи пізно вам доведеться звернутися до мови моделювання UML.

UML – це технологія створення схематичних діаграм. Програмісти використовують діаграми, щоб малювати класи, властивості і методи стандартними символами. Наприклад, класи (подібно кварталу – **Parcel**) зображені як прямокутники, властивості (подібно **Value** і **Zoning**) поряд із назвою мають спеціальні мітки і методи (подібно **CalculateTax**) поряд із назвами стрілок. Ці UML діаграми мають назву – діаграмами моделі об'єкта (рис. 7.1).

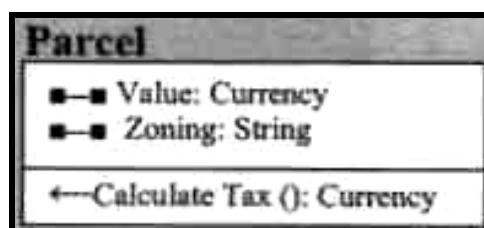


Рисунок 7.1 – UML діаграма моделі об'єкта

На діаграмі типи даних відображені зліва від властивостей і методів. Наприклад, ви бачите, що властивість **Zoning** має рядковий тип.

У мові VBA, щоб створити новий клас, формують новий модуль класу. Тобто скориставшись командою: **Insert / Class Module**, модуль можна зберегти в будь-якому з трьох проектів: поточному документі карти, новому або основному шаблоні. При цьому будуть розрізнятися рівні доступу до модулю, як і під час роботи з макросами та процедурами.

Щоб створити властивість для класу, необхідно оголосити перемінну з потрібним типом даних. Нехай клас **Parcel** буде мати дві властивості. Створіть їх, оголошуючи перемінні в модулі класу (рис. 7.2).

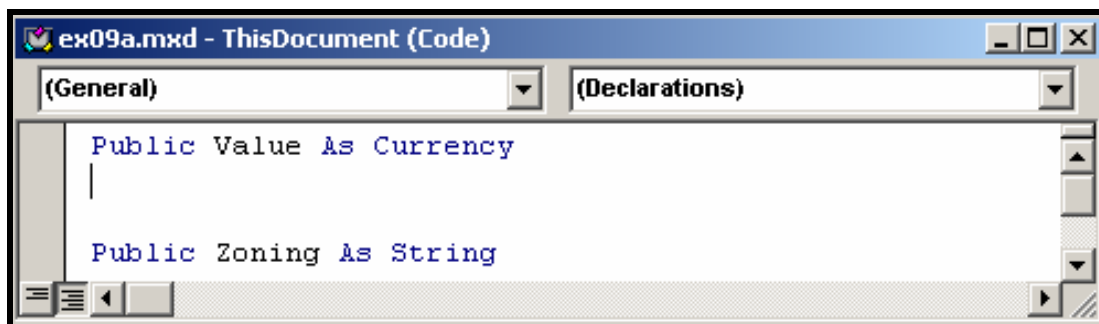


Рисунок 7.2 – Оголошення перемінних в модулі класу

Ці перемінні оголошуються поза процедурою з ключовим словом **Public**, замість ключового слова **Dim**. Перемінні, оголошені з оператором **Dim**, можуть використовуватися тільки в процедурі, в якій вони оголошені. Зверніть увагу, що перемінні **Value** та **Zoning** не належать певній процедурі.

Оголошення перемінних поза процедурою за допомогою оператора **Public** дозволяє використовувати їх в будь-якій процедурі та в будь-якому модулі коду (поки процедура знаходиться в поточному документі карти, оскільки клас зберігається в поточному документі карти).

Щоб створити метод для класу, необхідно закодувати підпрограму або функцію, залежно від того, чи повертає цей метод значення. Метод **CalculateTax** повертає розмір податку, тому даний метод був закодований як функція (код схожий на написаний раніше, але він не розміщується в обробнику події – клацання лівою кнопкою миші (ЛКМ) на кнопці форми). На діаграмі класу відзначено, що даний метод має тип **Currency** (грошовий). Очевидно, що він повертає значення вказаного типу. Аргументів він не має, тому дужки залишаються порожніми. Таким чином, оголошення даного методу буде виглядати наступним чином (рис. 7.3):

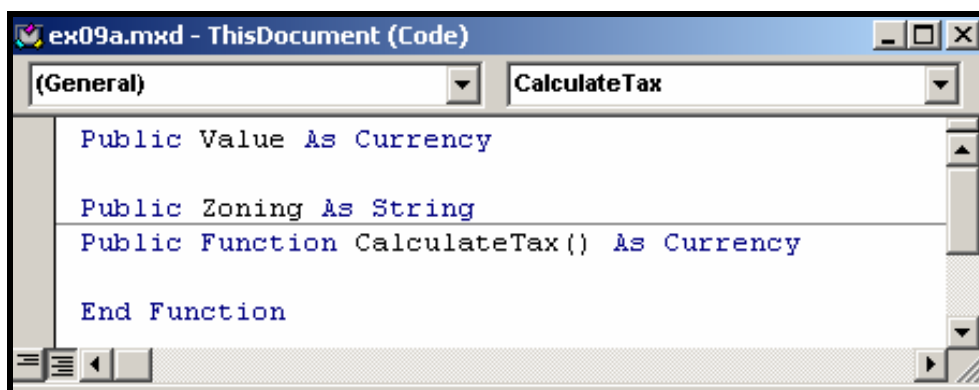


Рисунок 7.3 – Оголошення методу

Хід роботи

Вправа 1

Постановка задачі

Як програміст, який працює у проекті збереження живої природи (Ви працюєте з біологами, які спостерігають за життям слонів). В даний час, вчені використовують записники, щоб зберегти записи про слонів. Вони також роблять звукозаписи, як сурмить кожний слон в *.wav файл. Ваша задача: створити в ArcMap додаток для їхньої роботи. Оскільки об'єкта слон не існує в VBA або ArcGIS, треба створити новий клас для слона.

Біологи хочуть зберігати та працювати з іменами слонів і їхнім віком. Вони також хотіли б мати можливість прослуховувати ревіння слона. Вам необхідно використовувати цю інформацію, щоб створити діаграму UML класу **Слон (Elephant)** (рис. 7.4).

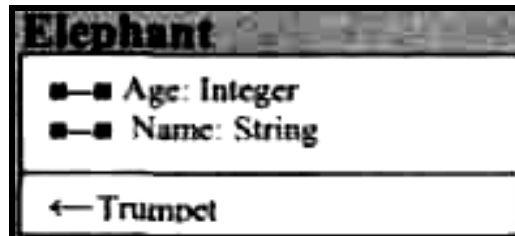


Рисунок 7.4 – діаграма UML класу **Слон**

В цій вправі, ви створите клас **Слон (Elephant)** з двома властивостями і методом, який запускає звуковий файл з ревом слона.

- Запустіть ArcMap і відкрийте вправу **ex09a.mxd**, розташовану в наступній папці: **...ArcObjects \ Chapter09**. Карта порожня, оскільки клас, який ви зараз створюєте не має географічної прив'язки.
- Увійдіть до редактора VBA.
- Вставте в проект модуль класу (**Insert / Class Module**).
- Змініть назву класу модуля, використовуючи вікно **Properties** (рис. 7.5):

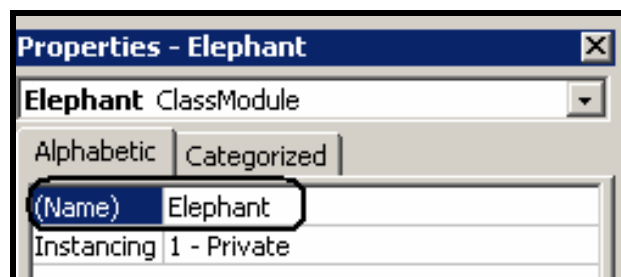


Рисунок 7.5 – Зміна назви класу модуля

- Опишіть властивості вашого класу у модулі класу наступним чином (рис. 7.6):

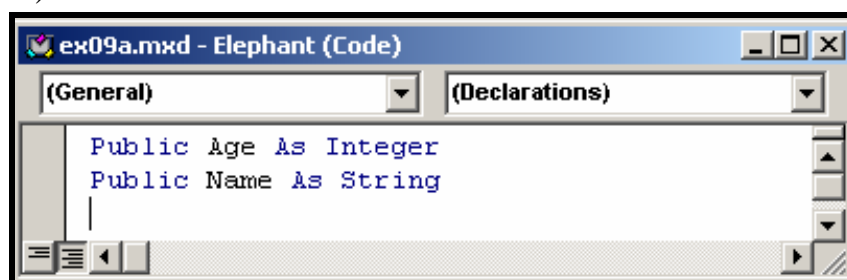


Рисунок 7.6 – Опис властивостей класу

- Додайте процедуру **Trumpet** (труба) за допомогою команди **Insert / Procedure** (рис. 7.7).
- Вставте усередину обрамляючих рядків процедури **Trumpet** наступні рядки (рис. 7.8):
- Шлях до звукового файлу в процедурі вкажіть за допомогою копіювання адресного рядка у Провіднику, заздалегідь знайшовши файл «**elephant.wav**». Цей код відрізняється від того, що зустрічався раніше.

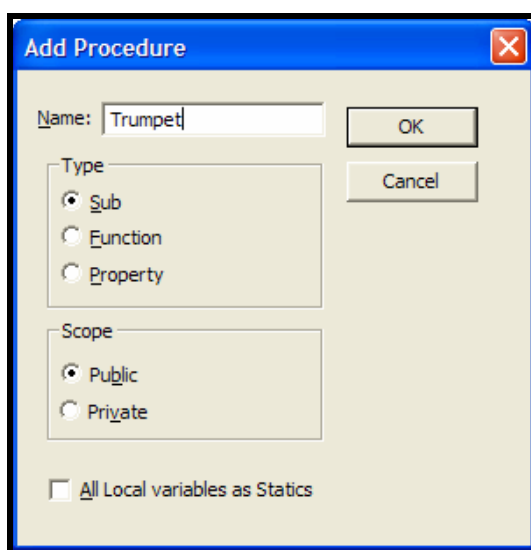


Рисунок 7.7 – Додавання процедури

Функція **sndPlaySound** не входить в VBA або ArcObjects. Вона є частиною **Microsoft Windows Application Programming Interface**, (програмного інтерфейсу додатків Microsoft Windows) або **API** функцією. **API** функції виконують завдання операційної системи подібно програванню звукових файлів, перевірки ваших прав доступу або пошуку шляху до папки **Temp**.

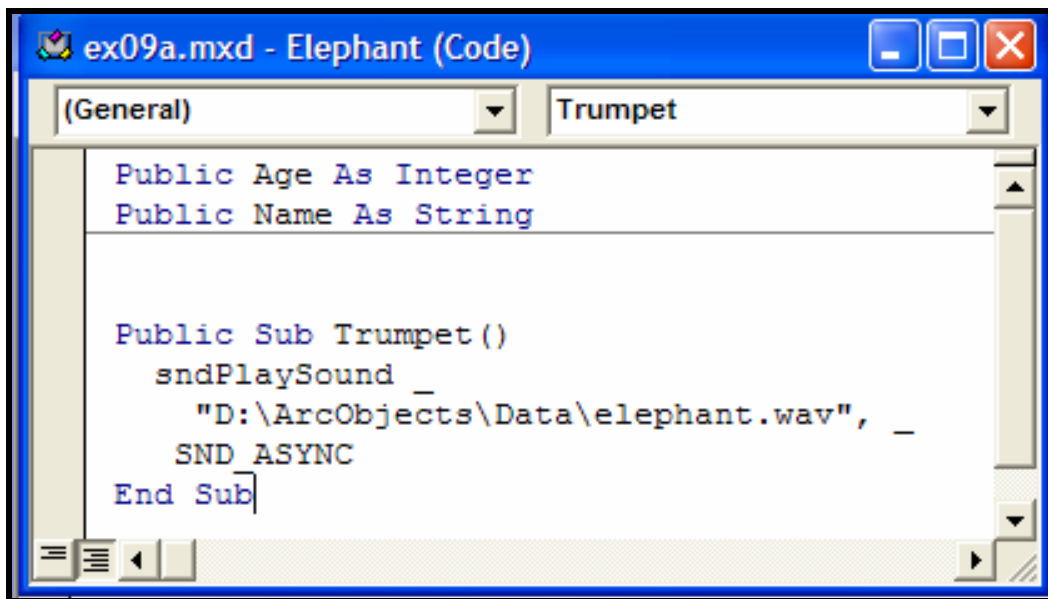


Рисунок 7.8 – Вікно коду

Щоб викликати функцію API в VBA, треба оголосити її в іншому модулі коду (ви робили це, коли оголошували функцію **KilogramToPound** в одному модулі, а викликали в іншому у попередній лабораторній роботі.) В цій вправі функція вже була оголошена в стандартному модулі **PlaySounds**.

ПРИМІТКА. Функція **sndPlaySound** має два параметри:

- перший – це шлях до звукового файлу ***.wav**;
- другий – константа, яка може бути або **SND_SYNC** або **SND_ASYNC**. Опція **SND_SYNC** робить паузу у виконанні коду, поки звуковий файл не закінчив програвання. Опція **SND_ASYNC** дозволяє виконувати код далі під час програвання звукового файлу.

Функції повертають значення. Але не завжди можна використовувати ці функції. Значення функції **sndPlaySound** – **Истина**, якщо звуковий файл програвся і **Ложь**, якщо цього не відбувається. Щоб не ускладнювати приклад, ми не будемо використовувати значення цієї функції.

Клас **Elephant** на цьому закінчений. Це – досить простий клас, але він використовує ті ж самі принципи кодування, які б Ви використовували для створення більш серйозних класів. Тепер ми можемо створювати об'єкт **Elephant**.

- Збережіть свою роботу.

Створення об'єктів

Тепер навчимося створювати об'єкти за допомогою призначеного для користувача інтерфейсу. Досі ви створювали форму шляхом вставки її в проект, а потім створювали об'єкт **CommandButton** шляхом перетягування його на форму.

Тепер ви будете створювати об'єкти за допомогою коду, оголошуючи і встановлюючи перемінні. Перемінні можуть бути не тільки даними основних типів, але і їх об'єктами.

Перемінні, які представляють дані основних типів називаються **вбудованими перемінними**, а ті які представляють об'єкти називаються **об'єктними перемінними**.

Оголошення і установка значень об'єктних перемінних дещо відрізняється від роботи з вбудованими перемінними. Для об'єктних перемінних замість типу вказується назва класу, а як оператор присвоєння використовується оператор **Set**. Але якщо Ви створюєте новий об'єкт, то використовується ключове слово **New**.

Dim E As Elephant

Set E = New Elephant

Тепер ви можете звертатися до властивостей і методів об'єкту, використовуючи перемінну **E**.

E.Name = "Mark" E.Trumpet

Потім, при бажанні, ви можете створити ще один об'єкт:

Dim E1 As Elephant

Dim E2 As Elephant

Set E1 = New Elephant

Set E2 = New Elephant

Після створення, ви можете звертатися до їхніх властивостей:

E1.Name = "Jerry"

E1.Age =24

E2.Name = "Ron"

E2.Age =28

Вправа 2

Постановка задачі

Для зручності біологів під час проведення досліджень необхідно створити об'єкт, за допомогою якого можна вводити будь-які характеристики спостережень.

У попередній вправі, Ви створили клас **Слон (Elephant)** з двома властивостями і методом, який запускає звуковий файл з ревом слона. В цій вправі ви скористаєтеся ним, щоб створити об'єкт «Слон».

- Запустіть ArcMap і відкрийте вправу **ex09b.mxd** розташовану у папці: ... **ArcObjects \ Chapter09**. На стандартній панелі інструментів ви побачите кнопку **AddElephant** із зображенням слона.

- Увійдіть до редактора процедури кнопки **AddElephant**. Ви побачите порожню заготовку, в якій необхідно написати код створення об'єкта **Слон (Elephant)**.

- На сторінці коду модулю **ThisDocument**, в події **AddElephant_click** необхідно додати наступні рядки (рис. 7.9):

Dim theElephant As Elephant
Set theElephant = New Elephant

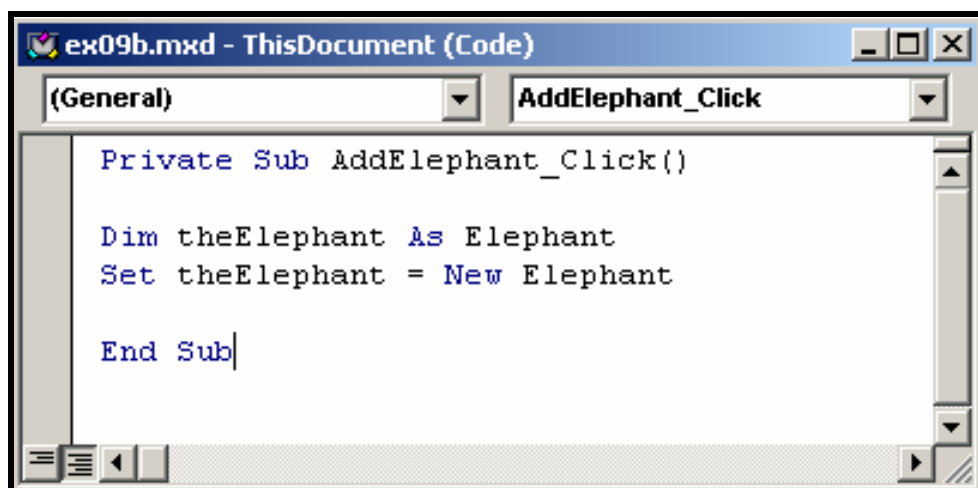


Рисунок 7.9 – Код події **AddElephant_click**

- Запустіть цю процедуру на виконання. Після цього в пам'яті комп'ютера виділиться місце під об'єкту перемінну, і перемінна **theElephant** буде їй відповідати.

- Продовжимо написання коду. Для цього знову увійдіть до обробника події **AddElephant_click**. Набірять наступні рядки, використовуючи підказки. Для цього набравши ім'я перемінної і поставивши крапку, почекайте поки VBA запропонує вам список доступних властивостей і методів даного об'єкта. Набірять наступне (рис. 7.10).

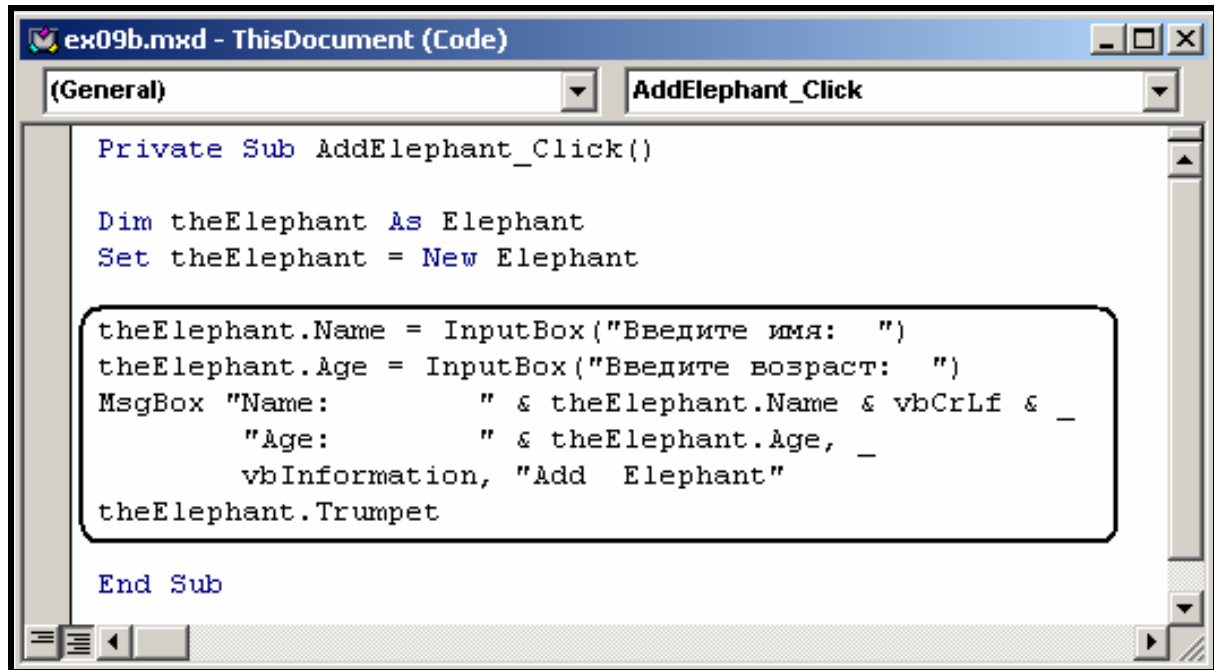


Рисунок 7.10 – Код процедури

- Закрийте редактор VBA і протестуйте кнопку. На запити, що з'явилися, відповідайте, наприклад: **Джек** (ім'я слона) та **35** (вік слона).

Після появи повідомлення про слона і натискання на кнопку **ОК** Ви повинні почути ревіння слона.

- Збережіть свою роботу.

Зверніть увагу на те, що у файлі **ex09b.mxd** модуль **PlaySound**, в якому є повідомлення для операційної системи вашого ПК, з якої бібліотеки потрібно підвантажити програвач музики (рис. 7.11).

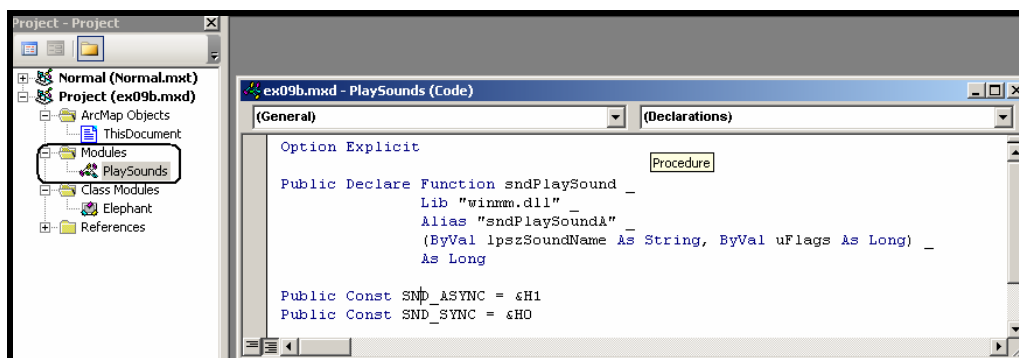


Рисунок 7.11 – Модуль для підвантаження програвача музики

Завдання для самостійного виконання

1. Розробіть свій власний клас відповідно до наочної області завдань по варіантах (табл. 1). В класі повинні бути як мінімум дві властивості і два методи.
2. * Напишіть програму, яка створює об'єкт вашого класу і виконує роботу з його властивостями і методами. Варіанти класів наведені у таблиці 7.1.
3. Створіть для цієї програми кнопку і протестуйте ваші розробки.

Таблиця 7.1 – Варіанти завдань на розробку класів

Варіант	Клас
1	Річка (River)
2	Дорога (Dear)
3	Станція (Station)
4	Житловий будинок (Dwelling-house)
5	Музей (Museum)
6	Гараж (Garage)
7	Дитячий садочок (Childs garden)
8	Школа (School)
9	Озеро (Lake)
10	Гора (Mountain)
11	Море (Iea)
12	Океан (Ocean)
13	Населений пункт (Zopulation item)
14	Адміністративна ділянка (Administrative site)
15	Спортивна команда (Command is sports)
16	Історичне місце (Historical place)

Контрольні питання

1. Сформулюйте своїми словами: для чого потрібно розробляти власні об'єкти.
2. Що таке UML?
3. Чим відрізняється клас від об'єкту.
4. Які ключові слова на мові VBA використовуються під час програмної роботи з об'єктами?
5. Чим відрізняється вбудовані і об'єктні перемінні?
6. Що таке функції API Windows?
7. Які параметри має функція sndPlaySound?

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений на ПК, роздрукований на аркушах формату А4. Звіт повинен містити (дивись лаб. роботу № 2). До звіту необхідно додати файл, виконаний у програмному забезпеченні ArcMap з виконаною роботою.

ЛАБОРАТОРНА РОБОТА № 8 АВТОМАТИЗАЦІЯ РОБОТИ В ARCMAP

Мета: виробити вміння і навички по вирішенню прикладних задач у програмі ArcMap за допомогою об'єктно-орієнтованої мови VBA.

Призначення: виконавши роботу, Ви навчитеся створювати обчислювальні процеси для вирішення прикладних завдань.

Підготовка до роботи

1. Які мови програмування можна використовувати у ArcGIS?
2. Для яких цілей при роботі у ArcMap використовують ArcObjects?
3. Для яких цілей використовують асинхронне виконання сервісів геообробки?
4. На якій мові програмування у ArcMap створюються скрипти?
5. Чи можна створений у одному файлі скрипт використати для вирішення інших задач геообробки даних?

Ключові слова

Скрипт, мова програмування, геообробка даних, об'єктно-орієнтоване програмування, сервіси.

Теоретичні відомості

В деяких ГІС-додатках є спеціальні набори інструментів для реалізації розширеної геоінформаційної бізнес-логіки. Ці інструменти можуть передбачити об'єм здобутої деревини в лісі, визначити відповідні ділянки для ресторану або оцінити зони розповсюдження токсичної хмари. Багато розробників для таких цілей використовують ArcObjects.

У багатьох випадках такі процеси можна описати в ArcGIS ModelBuilder, де їх графічно можна зв'язати у єдиний «ланцюжок». Такі моделі геообробки можна надавати як веб-сервіси і використовувати у веб-додатках. Переваги цього очевидні: використання сервісу геообробки до-

зволяє значно скоротити необхідний об'єм програмування. Крім того, можна використовувати переваги асинхронного виконання сервісів геообробки, чого непросто досягти у своєму коді ArcObjects.

Крім гнучкості, яка забезпечується наявністю сотень готових інструментів, які можна поєднувати в ModelBuilder, геообробка дає можливість розробляти власні інструменти. Найпростіший спосіб – це створення скриптів на мові Python, які можуть виконуватися самостійно або у поєднанні з іншими інструментами в моделі.

Щоб досягнути ще більшого контролю, замість Python для створення призначених для користувача інструментів геообробки, можна використовувати мови C#, Visual Basic, C++ або Java. Це дозволяє впроваджувати власну деталізовану логіку ArcObjects в моделі.

При використанні мови Python або іншої мови програмування перевага створення призначених для користувача інструментів полягає в тому, що їх можна використовувати повторно в інших робочих процесах, оскільки вони поведуться таким же чином, як і будь-який інший готовий інструмент. Крім того, код ArcObjects або Python може виконуватися в моделі асинхронного виконання сервісів геообробки, яка дуже зручна для довготривалих процесів.

Хід роботи

Вправа 1

Постановка задачі

Для зручності користувачів під час роботи з картами, які мають велику кількість шарів, Ви повинні створити процедури, які вимикають чи вмикають одразу усі шари.

- Запустіть ArcMap і відкрийте вправу **ex11a.mxd**, розташовану у наступній папці: **...ArcObjects \ Chapter11**.

- Введіть пароль **Carter**.

Вікно ArcMap покаже зображення, в якому Ви побачите вулиці Вашингтона округ Колумбія і декілька шарів з відмітками про злочини. На панелі інструментів **Crime Analysis** Ви бачите кнопку під назвою **ClearCrime**, яка вимкне усі шари.

- На панелі інструментів **Crime Analysis**, визвіть контекстне меню на кнопці **ClearCrime** і виберіть команду **View Source**. У відкритому вікні Visual Basic Editor Ви бачите **ClearCrime** – процедуру обробки подій. У цій процедурі необхідно написати наступний код (рис.8.1).
- Для тестування коду натисніть кнопку **ClearCrime**. У результаті усі шари повинні бути вимкнені (рис.8.2).
- Далі необхідно створити другу кнопку під назвою **AllCrime** (рис. 8.3).
- На панелі інструментів **Crime Analysis** визвіть контекстне меню на кнопці **AllCrime** та виберіть команду **View Source**. В результаті відкриється вікно Visual Basic Editor і Ви бачите процедуру обробки подій – **AllCrime**. У цій процедурі необхідно написати наступний код (рис. 8.4).
- В полі коду у властивості **Visible** для **pLayer**, яке встановлює видимість шарів, значення властивості необхідно змінити з **False** на **True**.

```

'процедура для виключення всіх шарів
Private Sub ClearCrime_Click()

    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument
    Dim pMap As IMap
    Set pMap = pMxDoc.FocusMap
    Dim pLayer As ILayer
    For i = 0 To pMap.LayerCount - 1
        Set pLayer = pMap.Layer(i)
        pLayer.Visible = False
    Next i
    pMxDoc.UpdateContents
    Dim pActiveView As IActiveView
    Set pActiveView = pMxDoc.ActiveView
    pActiveView.Refresh
End Sub

```

Рисунок 8.1 – Код процедури **ClearCrime**

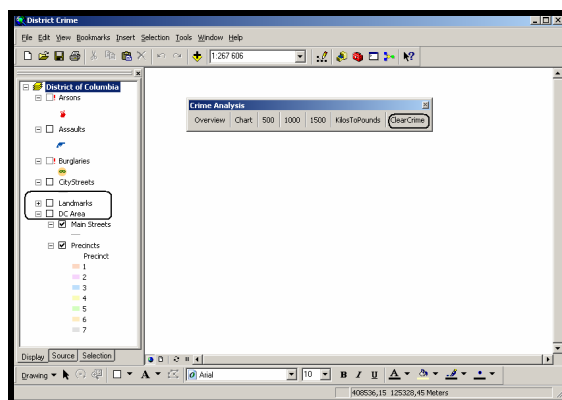


Рисунок 8.2 – Результат натискання кнопки **ClearCrime**

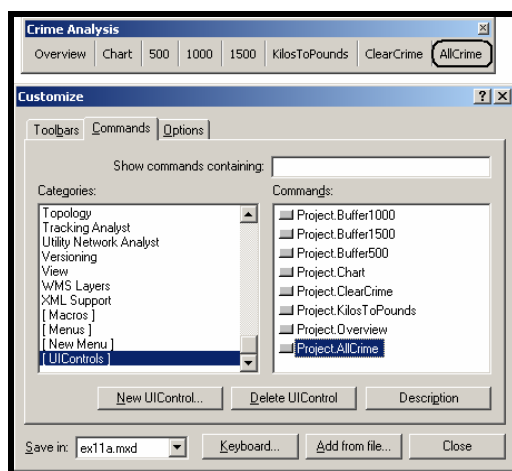


Рисунок 8.3 – Створення кнопки **AllCrime**

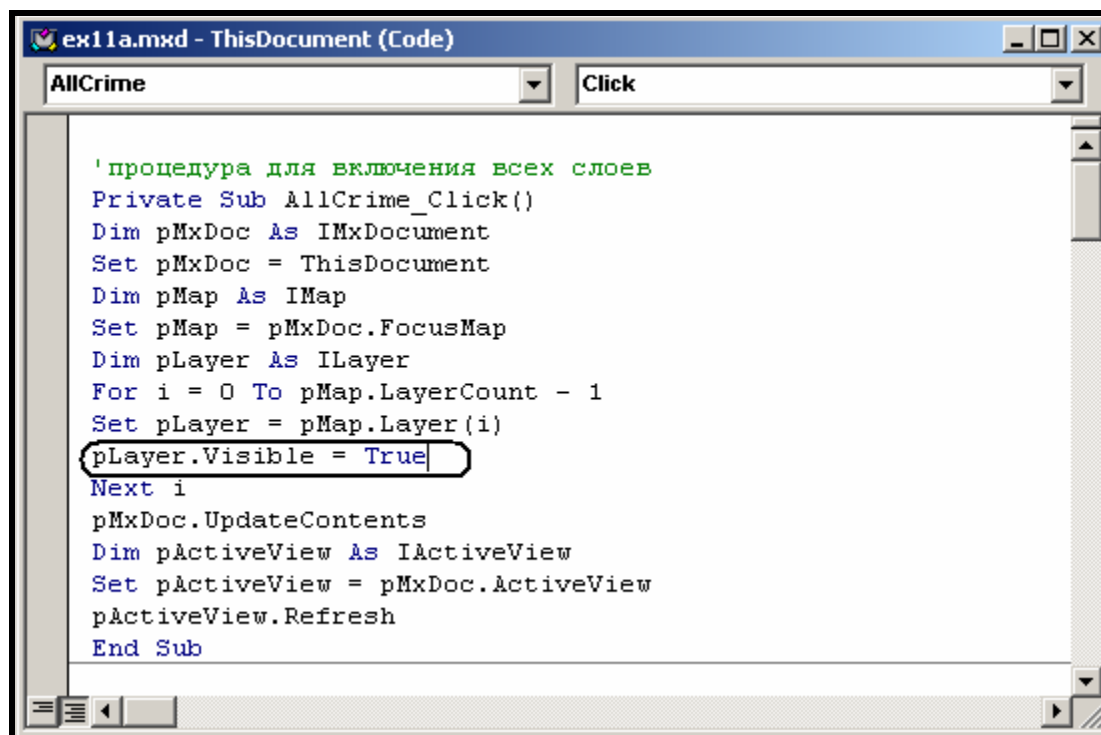


Рисунок 8.4 – Зміна значення властивості **pLayer**

- У результаті тестування коду ви переконаєтесь, що усі шари включені (рис. 8.5).

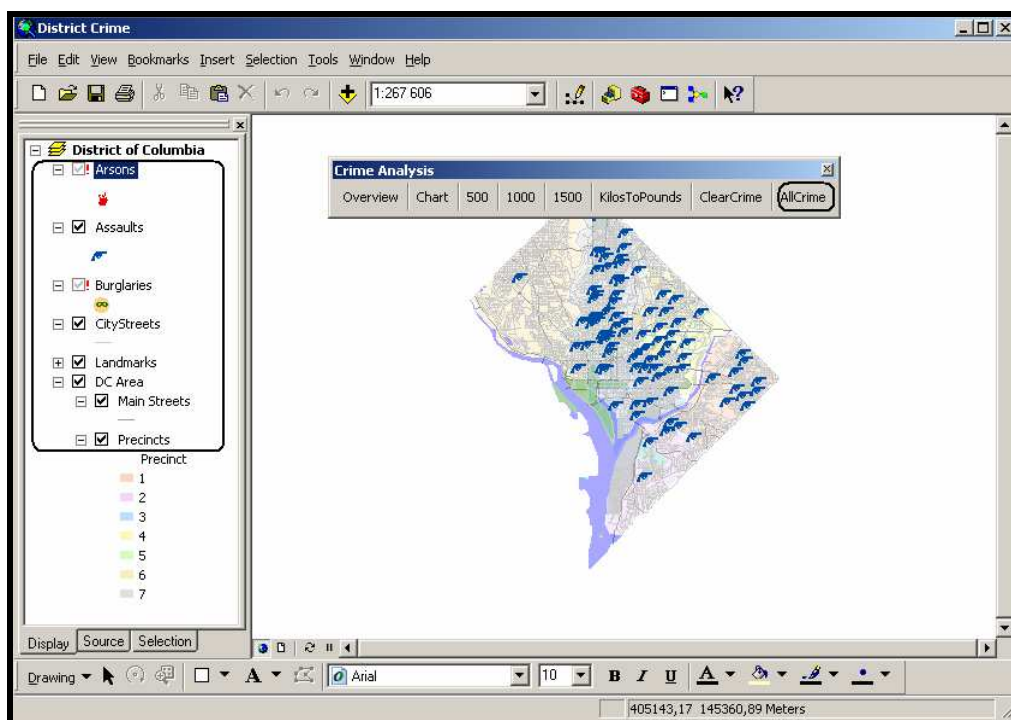


Рисунок 8.5 – Включені шари за допомогою зміни властивості

- Збережіть роботу.

Вправа 2

Постановка задачі

Для зручності користувачів під час роботи з картами та аналізу нанесених на них подій Ви повинні створити процедури, які змінюють колір сторінки карти.

- Запустіть ArcMap і відкрийте вправу **ex11b.mxd**, розташовану у папці: **...ArcObjects\Chapter11**.
- На панелі інструментів **Layout** створіть дві кнопки: **Page Color** і **BluePage**, додайте між ними роздільник для розміщення їх в окремих групах (рис. 8.6).

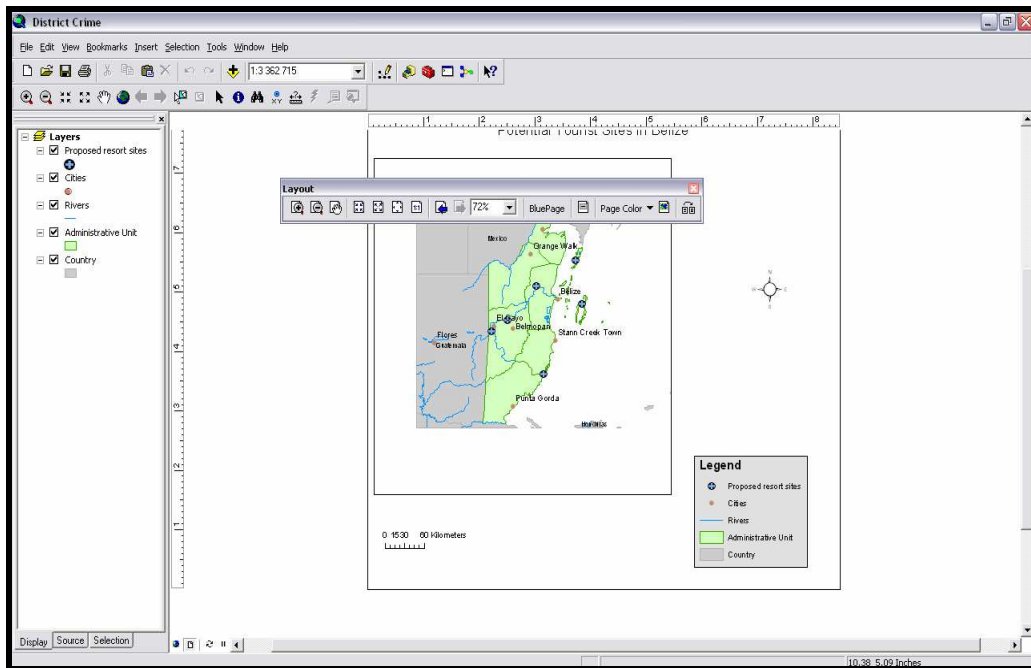


Рисунок 8.6 – Кнопки **Page Color** та **BluePage** в окремих групах

- Для кнопки **BluePage** необхідно створити наступний код (рис. 8.7).

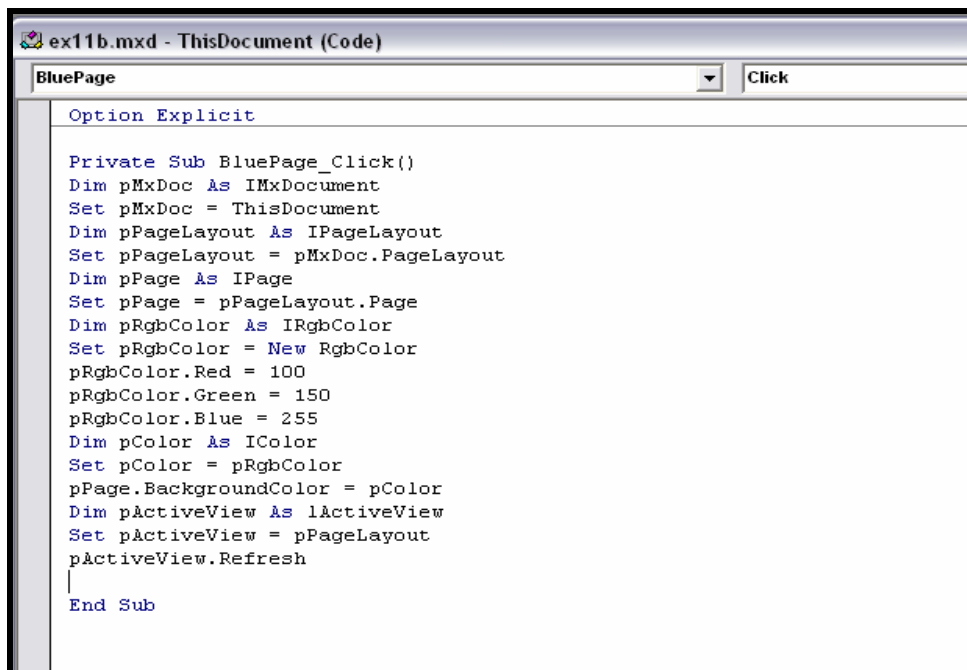


Рисунок 8.7 – Код для кнопки **BluePage**

- Закрийте Visual Basic Editor.
- В ArcMap, на панелі інструментів **Layout**, натисніть кнопку **BluePage**. Колір сторінки карти повинен зміниться на синій (рис. 8.8).

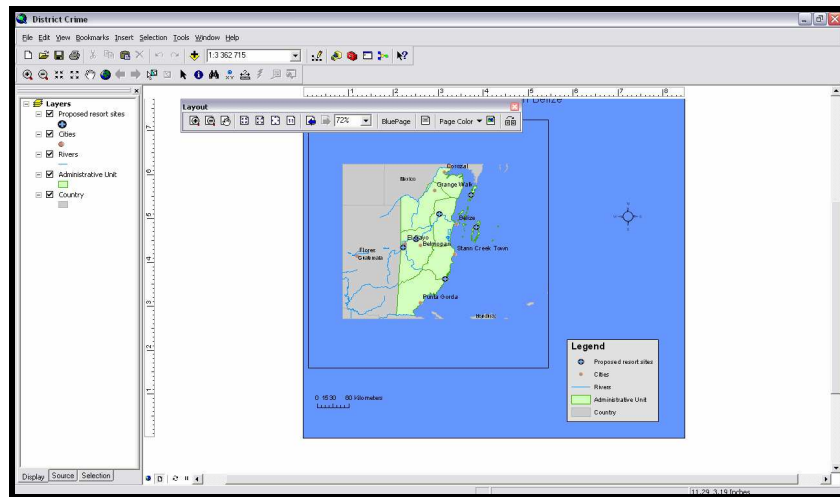


Рисунок 8.8 – Зміна кольору сторінки карти

- Для того, щоб колір сторінки змінився на білий, необхідно зробити іншу кнопку під назвою **WhitePage**.
- Прописуємо код для кнопки **WhitePage** аналогічно попередньому, але змінюємо його так:

pColor.Red = 255

pColor.Green = 255

pColor.Blue = 255

- Протестуйте кнопку

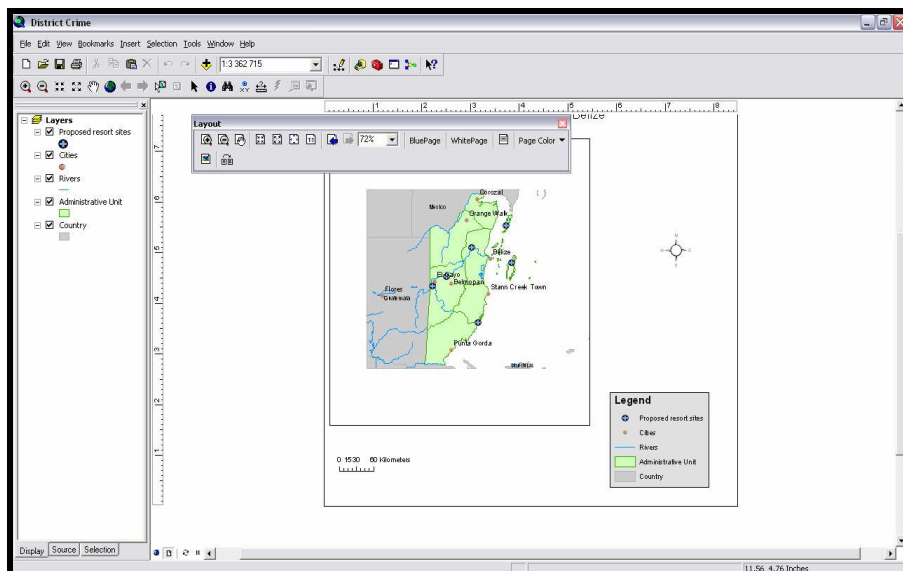


Рис. 8.9 – Результат тестування кнопки WhitePage

- Для області пустелі, в якій необхідно показати колір піску, робимо ще одну кнопку, називаємо її **SandPage** (рис. 8.10).

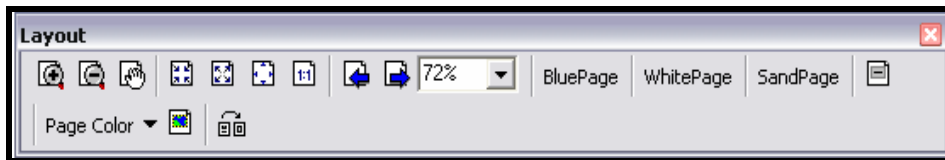


Рисунок 8.10 – Кнопки для зміни кольору сторінки карти

- Прописуємо код для кнопки **SandPage** аналогічно попередньому, але змінюємо наступне (рис. 8.11):

pColor.Red = 215
pColor.Green = 194
pColor.Blue = 158

```

Private Sub WhitePage_Click()
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pPageLayout As IPageLayout
Set pPageLayout = pMxDoc.PageLayout
Dim pPage As IPage
Set pPage = pPageLayout.Page
Dim pRgbColor As IRgbColor
Set pRgbColor = New RgbColor
pRgbColor.Red = 215
pRgbColor.Green = 194
pRgbColor.Blue = 158
Dim pColor As IColor
Set pColor = pRgbColor
pPage.BackgroundColor = pColor
Dim pActiveView As IActiveView
Set pActiveView = pPageLayout
pActiveView.Refresh
End Sub

```

Рисунок 8.11 – Код для кнопки **SandPage**

- Протестуйте кнопку та збережіть роботу.

Завдання для самостійного виконання

1. * Додайте кнопку на панель інструментів **Crime Analysis**, за допомогою якої можна вимикати один який-небудь шар на карті.
2. Додайте кнопку на панель інструментів **Layout**, натиснувши на яку, колір карти змінюється на жовтий.

Контрольні питання

Сформулюйте 5 контрольних питань відносно теми даної лабораторної роботи і дайте на них відповіді.

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи повинен бути оформлений на ПК, роздрукований на аркушах формату А4. Звіт повинен містити (дивись лаб. роботу № 2). До звіту необхідно додати файл, виконаний у програмному забезпеченні ArcMap з виконаною роботою.

РОЗРАХУНКОВО-ГРАФІЧНЕ ЗАВДАННЯ. РОЗРОБКА ГІС ПРОЕКТУ У СЕРЕДОВИЩІ ARCMAP ДЛЯ ВИРІШЕННЯ ПРИКЛАДНИХ ГІС ЗАДАЧ

У даному розділі розглядаються кроки виконання індивідуального розрахунково-графічного завдання (РГЗ) студента. Подані варіанти завдань та основні вимоги до створюваних проєктів.

Індивідуальне розрахунково-графічне завдання має на меті:

- систематизацію, закріплення, розширення теоретичних і практичних знань, умінь з питань розв'язання прикладних задач ГІС за допомогою VBA;
- використання набутих знань і вмінь під час реалізації конкретних геоінформаційних проєктів;
- набуття компетентності в галузі застосування технологій візуального програмування до розв'язання інформаційних задач у фаховій діяльності;
- розвиток навичок самостійної організації роботи під час виконанні РГЗ.

Актуальність даного виду роботи зумовлена розповсюдженістю проблем ефективної роботи з інформацією під час організації та повсякденному веденні справ геодезичного, картографічного характеру та справ з землеустрою.

Отже, розрахунково-графічне завдання є важливою частиною навчального модуля дисципліни і виконується студентами самостійно під керівництвом викладачів, які забезпечують викладання дисципліни.

Розрахунково-графічні завдання охоплюють широке коло різноманітних прикладних задач, які відповідають реальним ситуаціям і пов'язані єдиними підходами та методами для їхнього вирішення в середовищі ArcMap за допомогою VBA.

Основною метою РГЗ є формування у студентів основ професійного підходу до розв'язання інформаційних задач в майбутній фаховій діяльності: освоїти методику налаштування інтерфейсу користувача та створення власних проектів у середовищі ArcMap за допомогою мови програмування VBA.

Аналітична частина завдання полягає в постановці задачі та розробці інтерфейсу користувача, реалізованого за допомогою форм.

Дослідницька частина роботи полягає в тому, що студенти повинні самостійно з'ясувати необхідну додаткову інформацію стосовно заданої предметної області, а також повинні самостійно визначити коло допоміжних задач, пов'язаних з реалізацією конкретного проекту.

Проектна частина РГЗ передбачає безпосередню роботу в ArcMap і створення необхідних форм з об'єктами на них, з подальшим визначенням їх властивостей.

Загальні вимоги до оформлення РГЗ

Розрахунково-графічна робота повинна представляти собою звіт на листах книжкової орієнтації формату А4, оформлений на персональному комп'ютері і роздрукований. Звіт за індивідуальним завданням повинен містити:

- Титульний аркуш.
- Індивідуальне завдання, відповідно до номера в журналі групи.
- Розроблений приклад вирішення конкретної задачі засобами ArcMap, в який у свою чергу входять:
 - змістовна постановка задачі згідно свого варіанту;
 - блок-схеми алгоритму роботи процедур;
 - інтерфейс користувача, реалізований за допомогою форм, зовнішній вигляд якої визначається студентом самостійно;
 - об'єкти на формі повинні мати назви за Венгерською префіксною системою імен;
 - лістинги розв'язання з коментарями.

У разі необхідності характеристика використаних шарів у файлі: імена, типи, номери розташування до виконання, після виконання і т.п., залежно від розв'язуваної задачі.

Кожен студент оформляє звіт, роздруковує його та захищає. Результати роботи та захисту оцінюються згідно зі шкалою, яка наведена в таблиці 9.1.

Таблиця 9.1 – Шкала оцінювання РГЗ

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту, ІНДЗ	для заліку
90 – 100	A	відмінно	зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D	задовільно	
60 – 63	E		
35 – 59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0 – 34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Варіанти завдань

Згідно свого варіанту за студентським журналом розробити проект у ArcMap за допомогою Visual Basic for Applications.

ЗАВДАННЯ №1

Завдання А

1. Розробити форму для переведення введеного числа з одних одиниць виміру в інші (варіанти наведені в таблиці 9.2).
2. Вибрати одну з одиниць виміру початковою, і не менше трьох результатуючих.
3. На формі передбачити наступний набір елементів:
 - елемент для введення початкових одиниць;
 - елемент для вибору результатуючих одиниць або функцій обчислення з випадаючого списку;
 - елемент для представлення результату.
 Крім того, додати на форму дві кнопки:
 - кнопку для реалізації обчислень (ОБЧИСЛИТИ);
 - кнопку для закриття форми (ВИХІД).
4. Додати для форми відповідний малюнок.

5. Налаштувати властивості всіх доданих елементів, залежно від розробленого Вами дизайну форми.
6. Передбачте у кодї блокування кнопки «ОБЧИСЛИТИ» в разі введення до елемента для початкових даних текстових значень.
7. Передбачте у кодї появу різних малюнків на формі в разі розрахунку різних одиниць з випадаючого списку.
8. Створіть процедуру для запуску форми для переведення одиниць виміру разом з відкриттям документу ArcMap.
9. Створити кнопку для виклику форми на Вашій власній панелі інструментів. На кнопку додати напис – «Завдання А».

Таблиця 9.2 – Варіанти до завдання А

№ варіанту	Тема перерахунку
1	Калькулятор виміру нафти і газу
2	Калькулятор мір довжини: англійський – метричні
3	Калькулятор мір сили
4	Калькулятор мір площі
5	Калькулятор мір довжини: метричні – англійський
6	Обмін валют: внутрішньоєвропейські
7	Калькулятор мір потужності
8	Калькулятор мір енергії
9	Калькулятор мір об'єму
10	Обмін валют: міжконтинентальні
11	Тригонометричний калькулятор
12	Калькулятор мір швидкості
13	Калькулятор мір кутів
14	Калькулятор мір тиску
15	Калькулятор мір маси

Завдання Б

Розробити форму для обчислення процесу з розгалуженням (варіанти наведені в табл. 9.3).

1. На формі передбачити наступний набір елементів:
 - елемент для введення початкових одиниць;
 - елемент для представлення результату.

Крім того, додати на форму дві кнопки:

- кнопку для реалізації обчислень (ОБЧИСЛИТИ);
- кнопку для закриття форми (ВИХІД).

2. Додати для форми відповідний малюнок.

3. Налаштувати властивості усіх доданих елементів, залежно від розробленого Вами дизайну форми.

4. Створити кнопку для виклику вашої форми на власній панелі інструментів. На кнопку додати напис – «Завдання Б».

Таблиця 9.3 – Варіанти до завдання Б

№ варіанту	Задача
1	2
1	Визначити вартість товару, якщо відома роздрібна ціна і його кількість. При перевищенні деякої кількості одиниць товар продається за оптовою ціною (знижки на Р %)
2	У банку береться кредит на суму S грн., відсоток сплати за кредит залежить від терміну. Якщо термін кредиту не перевищує 3 місяці, то відсоток сплати складає Р1 %, 6 місяців - Р2 %, 1 рік - Р3 %, понад 1 рік - Р4 %. Яку суму треба буде повернути банку, якщо береться кредит строком на М місяців?
3	При купівлі автомобіля враховується його базова ціна (Б грн.), підвищена комфортність (К відсотків від базової ціни) і надбавка за фарбування кузова фарбою "металік" (М грн.). Визначити вартість автомобіля залежно від вимог покупця.
4	У супермаркеті з метою залучення покупців по вихідним дням (субота та неділя) встановлені знижки на 2 %. Визначити вартість покупки товару, якщо відома його роздрібна ціна, кількість та дата покупки.
5	У кінці дня в магазині підводяться підсумки, визначається виручка за день і порівнюється з середньоденною виручкою з початку поточного місяця. Визначити, чи був день вдалим, якщо відома поточна дата, загальна виручка за попередні дні місяця і виручка за поточний день.
6	Ціна на товар у фірмовому магазині перевищує собівартість на 2 %, а у всіх інших – на 5 %. Визначити ціну товару залежно від магазину, якщо відома його собівартість.
7	У трьох кінотеатрах міста показують різні фільми. Видати інформацію про те, який фільм демонструється у вибраному кінотеатрі і коли в ньому починаються сеанси.

Продовження таблиці 9.3

1	2
8	<p>Визначити заробітну платню робітника (ЗП), враховуючи його розряд за спеціальністю (Р) і стаж роботи (С).</p> $\text{ЗП} = \text{П} * \text{К}$ <p>де П - оклад робітника, грн.;</p> <p>К – коефіцієнт, який враховує розряд робітника і стаж його роботи (в роках)</p> $K = \begin{cases} 1, \text{ якщо } P * C \leq 8 \\ 1.3, \text{ якщо } 8 < P * C \leq 15 \\ 1.5, \text{ якщо } P * C > 15 \end{cases}$
9	<p>Підприємству встановлюється норма (Н) на споживання електроенергії. Визначити витрати підприємства Пл за користування електроенергією, якщо при дотриманні норми ціна за 1 квт. складає Ц1 грн. Якщо норма перевищена, але не більше ніж на 20 %, то за кожний наднормативний кіловат ціна складає Ц2 грн., якщо ж більше, ніж на 20 %, то додатково накладається штраф у розмірі N грн.</p>
10	<p>При почасовій оплаті роботи наднормова робота оплачується за подвоєною тарифною ставкою. Визначити заробіток робітника в день, якщо він працював у період від Час 1 до Час 2. Нормальна зміна триває 8 годин, почасова тарифна ставка складає Т грн.</p>
11	<p>Перевірка геодезичних приборів у компанії А перевищує базову вартість на 2 %, а у компанії Б – на 5 %. Визначити вартість перевірки залежно від компанії, яка проводить перевірку.</p>
12	<p>Визначити вартість проведення геодезичних вимірювань, якщо відома базова вартість. При перевищенні запланованого строку вимірювань робиться знижка (знижки на Р %)</p>
13	<p>При покупці геодезичного прибору враховується його базова ціна (Б грн.), додаткова комплектація (К відсотків від базової ціни) і надбавка за швидкість доставки (М грн.). Визначити вартість геодезичного прибору залежно від вимог покупця.</p>
14	<p>При проведенні геодезичних вимірювань з метою залучення замовників по вихідним дням (субота та неділя) встановлена знижка на 2 %. Визначити вартість проведенні вимірювань, якщо відома його базова ціна та дата проведення.</p>
15	<p>Під час здачі в оренду геодезичних приладів сплата за добу становить А грн. Сплата у вихідні та святкові дні становить А*К грн. Визначити сплату за оренду геодезичних приладів, якщо відомий час здачі та повернення приладів та наявність вихідних у цей проміжок часу.</p>

ЗАВДАННЯ № 2

Завдання В

Створити форму з текстовими полями атрибутів, в яких при виборі об'єкта на карті виводяться відповідні значення атрибутів. Створити кнопку для виклику вашої форми на власній панелі інструментів. На кнопку додати напис – «Завдання В1».

Сховати усі шари, які не є полігональними. Використовувати не менш ніж 10 шарів. В макросі використати цикли з постумовою. Створити кнопку для виклику вашого макросу на власній панелі інструментів. На кнопку додати напис – «Завдання В2». Підготувати другий макрос для послідовного перебору та включення усіх відключених шарів (тобто інвертування відключення). Створити кнопку для виклику вашого макросу на власній панелі інструментів. На кнопку додати напис – «Завдання В3».

Завдання Г

Розробити форму для обчислення циклічного процесу (варіанти наведені в табл. 9.4).

1. На формі передбачити наступний набір елементів:

- елемент для введення початкових одиниць;
- елемент для представлення результату.

Крім того, додати на форму дві кнопки:

- кнопку для реалізації обчислень (ОБЧИСЛИТИ);
- кнопку для закриття форми (ВИХІД).

2. Додати для форми відповідний рисунок.

3. Налаштувати властивості усіх доданих елементів, залежно від розробленого Вами дизайну форми.

4. Створити кнопку для виклику вашої форми на власній панелі інструментів. На кнопку додати напис – «Завдання Г».

Таблиця 9.4 – Варіанти до завдання Г

№ варіанту	Задача
1	2
1	В штатному розкладі підприємства є наступна інформація про N співробітників: ПІБ, посада, оклад. Знайти кількість співробітників, які обіймають задану посаду.


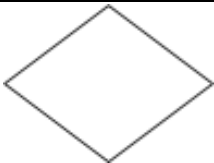





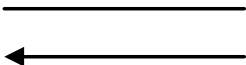
Продовження таблиці 9.4

1	2
2	На складі є N видів товарів, ціна кожного з яких відома. Знайти кількість товарів з ціною, більшою за задану
3	У штатному розкладі підприємства є наступна інформація про N співробітників: ПІБ, посада, оклад. Визначити посаду окремого співробітника.
4	На складі є N видів товарів, ціна і кількість кожного з яких відома. Визначити вартість усіх товарів.
5	Відомі результати стрибків у довжину N студентів. Знайти рекорд групи.
6	У штатному розкладі підприємства є наступна інформація про співробітників: ПІБ, посада, оклад. Знайти усіх співробітників з окладом менше заданої суми.
7	У штатному розкладі підприємства є наступна інформація про N співробітників: ПІБ, посада, оклад. Знайти середній оклад по підприємству. Знайти співробітників з окладом більше заданого.
8	На атестації студент отримав оцінки по N предмeтaх. Яка оцінка у нього найвища?
9	У штатному розкладі підприємства є наступна інформація про N співробітників: ПІБ, посада, оклад. Визначити оклад заданого співробітника.
10	На складі є N видів товарів, ціна кожного з яких відома. Знайти ціни товарів в іншій валюті, якщо заданий курс. Знайти товари, ціна яких менше за задану.
11	На складі є N видів товарів, ціна і кількість кожного з яких відома. Знайти товари, вартість яких більше за задану.
12	Відомі результати стрибків в довжину N студентів. Хто із студентів стрибнув далі за всіх?
13	На атестації студент отримав оцінки по N предмeтaх. Знайти предмети, по яких оцінка більше 70%.
14	У штатному розкладі підприємства є наступна інформація про N співробітників: ПІБ, посада, оклад. Знайти усі посади із заданим окладом.
15	Відомі ціни геодезичних приладів. Знайти найдорожчий прилад.



ДОДАТКИ

Додаток А

Таблиця А.1 – Основні символи схем алгоритмів

Назва	Графічне зображення	Функція символу
1	2	3
Процес		Виконання операції або групи операцій, в результаті яких змінюється значення, форма подання або розташування даних
Рішення		Вибір напрямку виконання алгоритму або програми, залежно від деяких змінних умов
Введення – виведення		Перетворення даних у форму, придатну для обробки (введення) або відображення результатів обробки (виведення)
Границі циклу		Відображення початку та закінчення циклу
Зумовлений процес		Використання раніше створених і окремо описаних алгоритмів або програм
Початок / закінчення		Початок, закінчення, переривання процесу обробки даних
Коментар		Зв'язок між елементом схеми та поясненням
Лінія потоку		Показник послідовності зв'язків між символами

Продовження таблиці А.1

1	2	3
З'єднувач в середині сторінки		Показчик зв'язку між перерваними лініями потоку, які зв'язують символи на одному аркуші
10. Міжсторінковий з'єднувач		Показчик зв'язку між роз'єднаними частинами схем алгоритмів і програм, розташованих на різних аркушах

Додаток Б

Перелік основних функцій

Таблиця Б.1 – Функції для роботи з датами

Функція	Призначення
Now	Поточні дата і час за комп'ютером
Date	Поточна дата за комп'ютером
Year(Дата)	Рік в аргументі Дата
Month(Дата)	Місяць в аргументі Дата
Day(Дата)	День в аргументі Дата
WeekDay(Дата)	Номер дня тижня в аргументі дата (неділі відповідає 1, а суботі – 7)
DateAdd(інтервал, кількість, Дата)	Нова дата, отримана додаванням до заданої дати кількості тимчасових інтервалів
DateDiff(інтервал, Дата1, Дата2)	Кількість часових інтервалів між першою і другою датами

Таблиця Б.2 – Допустимі значення аргументу «інтервал» у функціях DateAdd та DateDiff

Значення	Опис
yyyy	Рік
q	Квартал
m	Місяць
ww	Неділя
d	День
h	Година
n	Хвилина
s	Секунда

Таблиця Б.3 – Функції перетворення типів даних

Функція	Тип результату	Опис	Префікс
CBool (x)	Boolean	Логічне значення	bln
CByte (x)	Byte	Однобайтне ціле число (від 0 до 255)	byt
CInt (x)	Integer	Коротке ціле число (до $\pm 32\,000$)	int
CCur (x)	Currency	Число з фіксованою крапкою (грошовий тип)	cur
CDate (x)	Date	Дата та час	dtm
CDBl (x)	Double	Число з плаваючою крапкою подвійної точності (цілі, дробові 10 ± 348)	dbl
CLng (x)	Long	Довге ціле (± 2 млрд.)	lng
CSng (x)	Single	Число з плаваючою крапкою одинарної точності	sng
CStr (x)	String	Текстовий рядок	str
CVar (x)	Variant	Будь-яке значення з перерахованих вище	var

РЕКОМЕНДОВАНІ ДЖЕРЕЛА

Основні

1. Гурвиц Г. А. Microsoft Access 2010. Разработка приложений на реальном примере. – Изд.: BHV, 2010. – 496 с.
2. Exploring Arcobjects. ESRI, Redlands, 2004 г. 450 с.
3. ArcToolBox Руководство пользователя. – К.: ЗАО «ЕКОММ», 2003 г. 105 с.
4. ArcMap Руководство пользователя. – К.: ЗАО «ЕКОММ», 2003 г. 220 с.
5. ArcCatalog Руководство пользователя. – К.: ЗАО «ЕКОММ», 2003 г. 180 с.

Додаткові

6. Попов И.В., Чикинев М.А. Эффективное использование ArcObjects. Методическое руководство. – Новосибирск: СО РАН, 2003. – 160 с.

Ресурси мережі Internet

7. Сайт додатків Office корпорації Microsoft [Електронний ресурс]. – Режим доступу: <http://office.microsoft.com/ru-ru/access-help/>
8. Сайт корпорації «ArcGis Resource Center» по роботі з програмним продуктом – Режим доступу: <http://help.arcgis.com/>
9. Сайт «Высокие технологии». – Режим доступа: <http://www.citymap.odessa.ua>
10. Цифровий репозиторій ХНУМГ [Електронний ресурс]. – Режим доступу : <http://eprints.kname.edu.ua>

Навчальне видання

ПОМОРЦЕВА Олена Євгенівна

ПРОГРАМУВАННЯ ГЕОІНФОРМАЦІЙНИХ ЗАДАЧ

ЛАБОРАТОРНИЙ ПРАКТИКУМ

НАВЧАЛЬНИЙ ПОСІБНИК

Відповідальний за випуск: *О. Є. Поморцева*

За авторською редакцією

Комп'ютерний набір *О. Є. Поморцева*

Комп'ютерне верстання *І. В. Волосожарова*

Дизайн обкладинки *Д. О. Романова*

Підп. до друку 20.04.2015 р.

Друк на ризографі

Тираж 300 пр.

Формат 60x84/16

Ум. друк. арк. 6,6

Зам. №

Видавець і виготовлювач:

Харківський національний університет
міського господарства імені О. М. Бекетова,

вул. Революції, 12, Харків, 61002

Електронна адреса: rectorat@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК № 4705 від 28.03.2014 р.